

UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Matematica

Corso di Laurea Magistrale in Matematica

Tesi di Laurea Magistrale

Una nuova tecnica di cubatura quasi-Monte Carlo su domini 2D e 3D

Relatore

Prof. Stefano De Marchi

Correlatore

Prof. Alvisè Sommariva

Laureanda

Claudia Bittante

matr. 1034918

21 marzo 2014

A Sofa

Indice

Introduzione	vii
1 I metodi Monte Carlo e quasi-Monte Carlo	1
1.1 Il metodo Monte Carlo	2
1.1.1 Analisi dell'errore	3
1.1.2 Limiti del metodo Monte Carlo	4
1.2 I metodi quasi-Monte Carlo	4
1.2.1 Sequenze uniformemente distribuite	4
1.2.2 Discrepanza	6
1.2.3 Stime d'errore	9
1.2.4 Sequenza di Halton	10
1.2.5 Sequenza di Sobol	11
2 Metodi di cubatura	15
2.1 Polygauss	15
2.2 gqlens	19
2.3 Altri metodi di cubatura	21
3 La function lsqnonneg	25
3.1 L'algoritmo NNLS	25
3.2 Convergenza dell'algoritmo NNLS	28
3.3 L'implementazione	31
4 I punti approssimati di Fekete (AFP)	33
4.1 I punti di Fekete	34
4.2 Weakly Admissible Meshes (WAMs)	35
4.3 I punti approssimati di Fekete (AFP)	37
4.4 Cubatura sugli AFP	39
5 Esperimenti numerici su domini 2D	41
5.1 Poligono convesso	48
5.2 Poligono non convesso	56
5.3 Lente	61
5.4 Dominio composito	66
6 Esperimenti numerici su domini 3D	71
6.1 Cubo	74
6.2 Cono	79
6.3 Piramide	84

6.4 Dominio composito	89
Conclusioni	95

Introduzione

Il proposito della cubatura numerica consiste nell'approssimare l'integrale definito di una certa funzione continua $f : \Omega \subset \mathbb{R}^d \rightarrow \mathbb{R}$, relativamente a una funzione peso $w : \Omega \rightarrow \mathbb{R}$, diciamo $I(f) = \int_{\Omega} f(x)w(x)dx$ mediante una opportuna somma pesata $I_N(x) = \sum_{i=1}^N w_i f(x_i)$.

Nel caso univariato, supposto $\Omega = [a, b]$, in letteratura si è cercato di determinare formule che abbiano un basso numero di nodi, pesi w_i positivi (in virtù di alcune questioni di stabilità) e che risultino esatte per tutti i polinomi $p \in \mathbb{P}_n^d(\Omega)$ di grado totale inferiore o uguale a un certo intero n prefissato. Si è dimostrato che da questo punto di vista risultano ottimali le formule *gaussiane*, i cui nodi sono interni all'intervallo $[a, b]$, i pesi positivi e per cui $n = 2N - 1$. Tali formule sono pure *minimali*, cioè non esistono formule con meno di N nodi che permettano di essere esatte su tutti i polinomi di grado $n = 2N - 1$.

In ambito multivariato i risultati sono molto più limitati. Solo in domini particolari come triangolo, quadrato o disco si conoscono formule *minimali*, e la teoria per determinarli risulta alquanto complessa con risultati spesso espliciti solo per $n < 10$. Dal punto di vista pratico ci si accontenta spesso di determinare formule di cubatura $I_N(f)$, con un numero di nodi N anche maggiore della dimensione η_n dello spazio polinomiale $\mathbb{P}_n^d(\Omega)$. Ad esempio nel caso dei poligoni convessi o concavi con speciali proprietà, sono state recentemente implementate formule di cubatura a pesi positivi che sono esatte al grado $n = 2N - 1$ ma hanno un numero di nodi $N \approx LN^2$ maggiore di $\eta_{2N-1} = N(2N + 1)$.

I problemi peggiorano nel caso di domini con geometrie complicate, e la tipica strategia consiste nell'utilizzare metodi di tipo Monte Carlo o quasi-Monte Carlo, la cui convergenza all'integrale $I(f)$ da calcolare è tipicamente lenta, ma sfrutta il non irrilevante vantaggio che i relativi nodi sono punti casuali del dominio Ω e i pesi di facile determinazione.

In entrambi i casi ci si pone il problema di limitare la cardinalità N delle formule, così che sia almeno pari alla dimensione dello spazio polinomiale η_n , richiedendo comunque che i pesi siano positivi. Un primo passo in questo senso, ma solo in ambito univariato, è stato trattato da Huybrechs. In questo lavoro abbiamo considerato il più difficile caso multivariato.

Fissato un certo intero positivo n , una base $\{\phi_k\}_k$ di $\mathbb{P}_n^d(\Omega)$, e nota una formula di cubatura I_N su tale dominio, abbiamo cercato una nuova $I_{\bar{N}}$ a pesi positivi e nodi nel dominio Ω , con $\bar{N} \approx \eta_n$, tale che $S_{\bar{N}}(\phi_k) \approx I_N(\phi_k)$ per ogni k . In altri termini, la nuova formula ha circa gli stessi momenti della precedente, ma tipicamente un numero minore di nodi, peraltro positivi.

L'ingrediente fondamentale è risultato essere un algoritmo per problemi di minimi quadrati non negativi, implementata in Matlab mediante la routine

lsqnonneg. Se $V_{i,k} = \phi_k(x_i)$, con x_i nodi di una certa formula di cubatura, e $\gamma_k = I_N(\phi_k)$, tale routine, fissato n , calcola un vettore w avente circa η_n componenti non nulle (e peraltro positive) per cui $V^T w \approx \gamma$. Nel nostro caso, dal punto di vista algoritmico, ciò coincide con l'estrazione di una formula di cubatura a pesi positivi avente circa η_n punti e che calcoli i momenti quasi come la formula originaria. Come punti iniziali su cui applicare **lsqnonneg** si sono scelte nuvole di punti quasi-random sul dominio.

Un'alternativa è quella dei *punti approssimati di Fekete (AFP)*, valida per i domini compatti. Si tratta di un insieme di punti estratti da mesh di punti che discretizzano il dominio, e che su tale discretizzazione massimizzano, in valore assoluto, il determinante di Vandermonde del dominio. Il calcolo degli AFP avviene tramite la semplice fattorizzazione QR con pivoting per colonne mediante la routine di Matlab **approxfek**. Nel caso di *Weakly Admissible Meshes (WAMs)*, ovvero griglie di punti quasi-ottimali per l'approssimazione ai minimi quadrati, si è visto che gli AFP hanno distribuzione asintoticamente uguale ai punti di Fekete. In questo lavoro si è scelto di estrarre gli AFP da un insieme di punti quasi-random sul dominio.

Nel capitolo numerico abbiamo studiato il comportamento numerico di tali approcci su problemi bivariati e trivariati, sia in domini in cui è nota una formula di cubatura, ma la cui cardinalità N è eccessiva, sia in domini in cui la geometria era così complessa da richiedere l'uso di un metodo di tipo quasi-Monte Carlo con un numero eccessivo di punti, di seguito ridotto dall'algoritmo.

I codici Matlab utilizzati sono disponibili nel CD allegato al lavoro.

Capitolo 1

I metodi Monte Carlo e quasi-Monte Carlo

Il metodo Monte Carlo, scoperto negli anni '40 e basato sul campionamento in nodi casuali, venne inizialmente concepito per l'integrazione numerica su dimensioni $d \geq 2$, ma le sue applicazioni vengono usate nei campi più disparati.

La ragione del suo successo risiede nel fatto che, per primo, fornì un metodo risolutivo per l'integrazione numerica, *indipendente dalla dimensione* dello spazio. Questa proprietà risulta di particolare importanza, in quanto i metodi classici si basavano su prodotti cartesiani di regole di integrazione 1-dimensionali, e garantivano un bound d'errore dell'ordine $\mathcal{O}(N^{-2/d})$, dove N e d sono rispettivamente il numero di campioni e la dimensione dello spazio. È evidente che la stima dell'errore peggiora al crescere della dimensione, richiedendo di aumentare esponenzialmente con d il numero di campioni N , e rendendo, quindi, impraticabile la risoluzione al calcolatore.

Il fatto di utilizzare campionamenti random, tuttavia, fa sì che il metodo Monte Carlo abbia un'impronta fortemente stocastica, e questo si riflette sul tipo di stime d'errore che fornisce, ovvero probabilistiche (per l'appunto "stime", anziché veri e propri "bound"). Inoltre, il metodo Monte Carlo deve servirsi di subroutines deterministiche per generare numeri random al calcolatore, che vengono perciò detti *pseudo-random*; uno dei problemi che si sono dovuti affrontare, quindi, è stato quello di creare algoritmi che generino campioni "sufficientemente" random, ovvero che mimino alcune precise proprietà di casualità. D'altro canto, si è visto che una scelta perfettamente random dei campioni può essere controproducente per la convergenza, poiché in molti problemi come l'integrazione numerica questa è migliore se i campioni sono distribuiti con una certa uniformità; cosa su cui invece i campionamenti random non possono contare, dal momento che generalmente contengono addensamenti di punti.

Per tutti questi motivi, sono stati introdotti metodi quasi-Monte Carlo, che possono essere considerati la versione deterministica del metodo Monte Carlo, in quanto utilizzano punti deterministici "ben scelti" anziché punti pseudo-random (le cosiddette *sequenze quasi-random* o *a bassa discrepanza*), e forniscono un bound d'errore deterministico che è migliore della stima probabilistica data dal metodo Monte Carlo.

In questo capitolo verranno presentati i metodi Monte Carlo e quasi-Monte

Carlo per l'integrazione numerica; verrà introdotto il concetto di discrepanza, necessario per la creazione delle sequenze quasi-random; e saranno riportati i risultati noti per le convergenze dei due metodi. Per una trattazione più approfondita dell'argomento, si rimanda a [48, 44, 9].

1.1 Il metodo Monte Carlo

Consideriamo il seguente problema di integrazione: vogliamo calcolare l'integrale $I(f) := \int_{\Omega} f(x)dx$, dove $f : \Omega \rightarrow \mathbb{R}$ e $\Omega \subseteq \mathbb{R}^d$ ha misura di Lebesgue d -dimensionale finita $0 < \lambda_d(\Omega) < \infty$. L'approccio del metodo Monte Carlo è quello di interpretare questo integrale in modo stocastico, esprimendolo come prodotto del valore medio di f con la misura del dominio [48, 9]. A tal fine, consideriamo il dominio di integrazione Ω come uno spazio di probabilità dotato di misura di probabilità $d\mu = dx/\lambda_d(\Omega)$. Allora, per $f \in L^1(\mu)$, si ha

$$\int_{\Omega} f(x)dx = \lambda_d(\Omega) \int_{\Omega} f(x)d\mu(x) = \lambda_d(\Omega)E(f) \quad (1.1)$$

dove $E(f)$ è il *valore atteso* [43] della variabile aleatoria f . In questo modo il problema del calcolo dell'integrale di f è ridotto a quello di determinare il valor medio di f .

Per calcolare il valore atteso di f si ricorre ad un'escamotage che viene dalla statistica, ossia si effettua una media ponderata dei valori assunti da f su un numero arbitrario (e finito) di punti random del dominio. Ovvero, presi N punti $x_1, \dots, x_N \in \Omega$ indipendenti e distribuiti uniformemente rispetto alla probabilità μ , si costruisce la *stima di Monte Carlo* del valore atteso $E(f)$

$$E(f) \approx E_N(f) := \frac{1}{N} \sum_{i=1}^N f(x_i). \quad (1.2)$$

La (1.2) trova la sua giustificazione teorica nella Legge Forte dei Grandi Numeri [43], poiché

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i) = E(f). \quad (1.3)$$

Ora, applicando la stima (1.2) alla formula (1.1), si ottiene la seguente *stima di Monte Carlo* per l'integrale di f , che è quella che si va effettivamente ad usare nell'integrazione numerica Monte Carlo:

$$I(f) = \int_{\Omega} f(x)dx \approx \frac{\lambda_d(\Omega)}{N} \sum_{i=1}^N f(x_i) =: I_N(f) \quad (1.4)$$

Poiché si tratta di una stima di tipo probabilistico, per garantire un risultato statisticamente più fedele a quello reale cercato, in genere si esegue per più volte il calcolo dell'integrale con il metodo Monte Carlo, cambiando gli N punti random ad ogni esecuzione dell'algoritmo, e considerando infine la media dei valori $I_N(f)$ ottenuti.

1.1.1 Analisi dell'errore

Definiamo l'errore dell'integrazione Monte Carlo

$$\epsilon_N := I(f) - I_N(f). \quad (1.5)$$

La *varianza* [43] di f è

$$\sigma^2(f) = \int_{\Omega} (f(x) - E(f))^2 d\mu(x), \quad (1.6)$$

ed è finita se $f \in L^2(\mu)$.

Vale il seguente teorema di teoria della probabilità (per la dimostrazione, cfr. Theorem 1.1 [48]).

Teorema 1. *Sia $f \in L^2(\mu)$. Allora, per ogni $N \geq 1$, si ha*

$$\int_{\Omega} \cdots \int_{\Omega} \left(\frac{1}{N} \sum_{i=1}^N f(x_i) - E(f) \right)^2 d\mu(x_1) \cdots d\mu(x_N) = \frac{\sigma^2(f)}{N}. \quad (1.7)$$

Questo risultato dice che l'errore commesso nell'approssimare il valore atteso (1.2) è, in media, $\sigma(f)N^{-1/2}$, dove $\sigma(f) = (\sigma^2(f))^{1/2}$ è la *deviazione standard* [43] di f .

Corollario 2. *Sotto le ipotesi del Teorema 1, si ha la seguente stima dell'errore dell'integrazione Monte Carlo (1.4)*

$$E(\epsilon_N^2)^{1/2} = \lambda_d(\Omega) \sigma(f) N^{-1/2}. \quad (1.8)$$

Un'altra informazione sull'errore viene dal Teorema del Limite Centrale [43], una cui riformulazione adatta al nostro caso è la seguente.

Teorema 3. (Teorema del Limite Centrale). *Sia $\sigma^2(f)$ la varianza di f , tale che $0 < \sigma^2(f) < \infty$. Allora esiste una variabile normale standard $\nu \in N(0, 1)$ tale che*

$$\lim_{N \rightarrow \infty} \epsilon_N(f) = \lambda_d(\Omega) \sigma(f) N^{-1/2} \nu. \quad (1.9)$$

Ciò significa che l'errore dell'integrazione Monte Carlo è dell'ordine $\mathcal{O}(N^{-1/2})$, con una costante moltiplicativa che è la radice quadrata della varianza della funzione integranda, moltiplicata per l'area del dominio. Inoltre, grazie a questo teorema sappiamo che la distribuzione dell'errore è approssimativamente quella di una variabile aleatoria normale. Si noti la peculiarità di questo risultato: anziché un upper bound, abbiamo una stima probabilistica della grandezza dell'errore e della sua distribuzione.

Questa stima, tuttavia, ha il notevole pregio di essere indipendente dalla dimensione d dello spazio, a differenza dei bound d'errore delle regole d'integrazione d -dimensionale classiche, che sono invece dell'ordine $\mathcal{O}(N^{-2/d})$. Da un punto di vista computazionale, l'esperienza mostra che il metodo Monte Carlo è sempre preferibile a quelli classici già per dimensioni $d \geq 5$.

1.1.2 Limiti del metodo Monte Carlo

Nonostante il fatto di essere indipendente dalla dimensione dello spazio, il metodo Monte Carlo presenta alcuni punti a sfavore che è bene tener presenti.

Per prima cosa, il Corollario 2 ed il Teorema 3 forniscono soltanto stime probabilistiche dell'errore di integrazione, anziché upper bound. D'altronde, non ci si potrebbe aspettare diversamente, data la natura probabilistica della formula stessa di integrazione (1.4).

Come seconda cosa, si richiede soltanto l'ipotesi che la funzione integranda sia $L^2(\mu)$. Tuttavia, a differenza di quanto accade generalmente nell'analisi numerica, la regolarità della funzione integranda *non* incide sulla bontà dell'approssimazione.

Infine, generare campioni random non è una questione semplice; questo problema viene solitamente "risolto" ricorrendo a campioni pseudo-random anziché random, ovvero generati tramite algoritmi deterministici e che mimino alcune proprietà dei punti random.

Per migliorare le prestazioni del metodo Monte Carlo esistono attualmente due diversi approcci. Il primo è quello di ridurre la varianza, e comprende tecniche come il campionamento stratificato (*stratified sampling*), il metodo delle variabili antitetiche (*antithetic variates*), quello della variabile di controllo (*control variates*), la tecnica del matching tra momenti (*matching moments method*) e quella dell'*importance sampling* [9]. Il secondo approccio, invece, è quello di migliorare l'esponente $1/2$ della stima d'errore (1.9), andando ad agire sulle probabilità che concorrono a determinare tale stima: questa è la strada percorsa dai metodi quasi-Monte Carlo, che si distinguono dal metodo Monte Carlo proprio per la scelta diversa, non più random, dei campioni.

1.2 I metodi quasi-Monte Carlo

Come già accennato nella sezione precedente, l'idea di base per i metodi quasi-Monte Carlo è quella di sostituire i campioni random del metodo Monte Carlo con punti deterministici opportunamente scelti. Il criterio per la loro scelta dipende dal problema da risolvere. Nel caso dell'integrazione numerica, una scelta comune cerca *sequenze uniformemente distribuite e/o a bassa discrepanza* [48, 44].

1.2.1 Sequenze uniformemente distribuite

Supponiamo di avere come dominio di integrazione Ω il cubo unitario d -dimensionale chiuso $[0, 1]^d$. Data la funzione integranda $f : [0, 1]^d \rightarrow \mathbb{R}$, si consideri l'approssimazione quasi-Monte Carlo

$$\int_{[0,1]^d} f(x)dx \approx \frac{1}{N} \sum_{i=1}^N f(x_i), \quad (1.10)$$

dove $x_1, \dots, x_N \in [0, 1]^d$. Idealizzando il modello, possiamo supporre di avere una sequenza infinita $\{x_i\}_{i \in \mathbb{N}} \subset [0, 1]^d$, invece della sequenza finita x_1, \dots, x_N . Ovviamente, bisogna richiedere che con tale sequenza si ottenga dalla (1.10) un

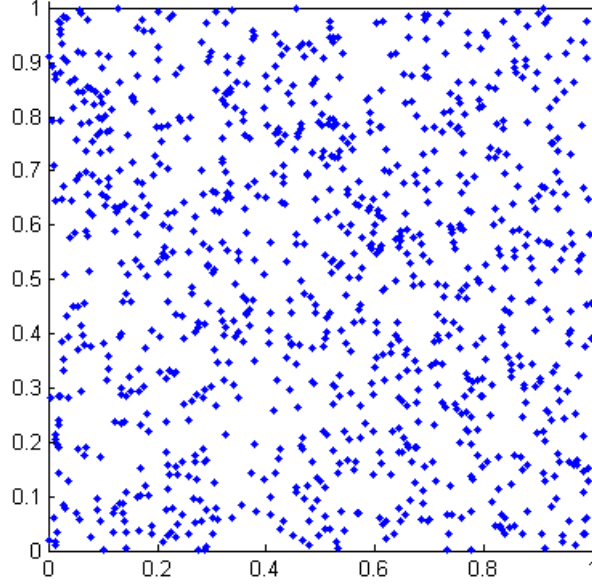


Figura 1.1: Punti random su $[-1, 1]^2$, $N = 1000$.

metodo convergente, ovvero si vuole una sequenza $\{x_i\}_{i \in \mathbb{N}}$ tale che

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i) = \int_{[0,1]^d} f(x) dx \quad (1.11)$$

valga per una classe ragionevolmente ampia di funzioni integrande, come ad esempio quella delle funzioni in $\mathcal{C}([0, 1]^d)$. D'altra parte, la condizione (1.11) equivale a richiedere che la sequenza $\{x_i\}_{i \in \mathbb{N}}$ sia *uniformemente distribuita* su $[0, 1]^d$ [44].

Definizione 1. Siano E uno spazio di Hausdorff compatto e μ una misura positiva regolare normata di Borel [38], ovvero una misura positiva μ così definita sui boreliani di E :

- (i) $\mu(E) = 1$,
- (ii) $\mu(F) = \sup\{\mu(C) : C \subseteq F, C \text{ chiuso}\}$ per ogni boreliano F in E ,
- (iii) $\mu(F) = \inf\{\mu(D) : F \subseteq D, D \text{ aperto}\}$ per ogni boreliano F in E .

La sequenza $\{x_i\}_{i \in \mathbb{N}} \subset E$ è detta **μ -uniformemente distribuita in E** se per ogni $f \in \mathcal{C}(E)$ vale

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N f(x_i) = \int_E f(x) d\mu(x), \quad (1.12)$$

dove con $\mathcal{C}(E)$ indichiamo lo spazio delle funzioni continue.

Si può dimostrare che per ogni arbitrario spazio di Hausdorff compatto E e per ogni misura positiva regolare normata di Borel μ , esiste una sequenza $\{x_i\}_{i \in \mathbb{N}} \subset E$ μ -uniformemente distribuita in E [44]. Si dimostra che è equivalente alla Definizione 1 la seguente

Definizione 2. Siano E uno spazio di Hausdorff compatto e μ una misura positiva regolare normata di Borel. La sequenza $\{x_i\}_{i \in \mathbb{N}} \subset E$ è detta **μ -uniformemente distribuita in E** se per ogni sottoinsieme $B \subset E$ vale

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N \chi_B(x_i) = \mu(B), \quad (1.13)$$

dove χ_B è la funzione caratteristica di B .

Per semplicità, nel seguito della trattazione assumeremo $\mu = \lambda_d$, ovvero la misura di Lebesgue d -dimensionale, in accordo con [48].

Quanto detto lascia intuire che i campioni da usare nella formula di integrazione quasi-Monte Carlo (1.10) debbano avere una distribuzione simile a quella uniforme su $[0, 1]^d$, ovvero è auspicabile che i punti siano distribuiti in modo omogeneo su $[0, 1]^d$. Questo comportamento non è in generale riscontrabile nei punti random, che, avendo distribuzione casuale, tendono inevitabilmente ad addensarsi, come si può vedere nella figura 1.1.

1.2.2 Discrepanza

Il concetto di discrepanza nasce dalla necessità di misurare in qualche modo il grado di distribuzione uniforme di un insieme di punti. Esistono diverse definizioni di discrepanza, ma il punto in comune è che forniscono una misura dell'irregolarità della distribuzione di un insieme di punti [48, 44].

Siano J un insieme e $X = \{x_1, \dots, x_N\} \subset J$ una sequenza finita di punti di J . Preso un arbitrario sottoinsieme B di J , definiamo la *counting function* $\#(B, X)$ che conta il numero di indici j per i quali $x_j \in B$, $1 \leq j \leq N$, ovvero

$$\#(B, X) = \sum_{j=1}^N \chi_B(x_j), \quad (1.14)$$

dove χ_B è la funzione caratteristica di B .

Definizione 3. Sia $J = \prod_{i=1}^d [\alpha_i, \beta_i) = \{x \in \mathbb{R}^d : \alpha_i \leq x_i < \beta_i\}$. Sia $X = \{x_1, \dots, x_N\} \subset J$ una sequenza finita di punti di J . Si definisce **discrepanza dell'insieme di punti X** la quantità

$$D(X) = \sup_{B \subseteq J} \left| \frac{\#(B, X)}{N} - \lambda_d(B) \right|, \quad (1.15)$$

dove i sottoinsiemi B di J sono della forma $B = \{y \in J : a_i \leq y_i < b_i, i = 1, \dots, d\}$.

Si noti che necessariamente $0 \leq D(X) \leq 1$.

Nel caso in cui l'insieme J della Definizione 3 abbia un vertice in 0, si usa la definizione di *discrepanza stellata*.

Definizione 4. Sia $J^* = \prod_{i=1}^d [0, \beta_i) = \{x \in \mathbb{R}^d : 0 \leq x_i < \beta_i\}$. Sia $X = \{x_1, \dots, x_N\} \subset J^*$ una sequenza finita di punti di J^* . Si definisce **discrepanza stellata dell'insieme di punti X** la quantità

$$D^*(X) = \sup_{B \subseteq J^*} \left| \frac{\#(B, X)}{N} - \lambda_d(B) \right|, \quad (1.16)$$

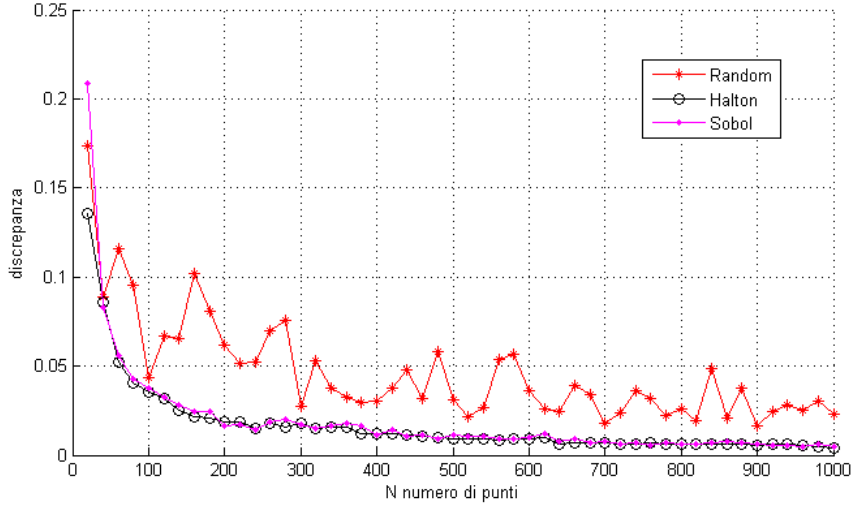


Figura 1.2: Valori della discrepanza stellata dei punti random, di Halton e di Sobol su $[0, 1]^2$, al variare del numero di punti N .

dove i sottoinsiemi B di J^* sono della forma $B = \{y \in J^* : 0 \leq y_i < b_i, i = 1, \dots, d\}$.

La discrepanza e la discrepanza stellata sono legate dalle seguenti relazioni [44].

$$D^*(X) \leq D(X) \leq 2^d D^*(X). \quad (1.17)$$

Nel caso più generico in cui l'insieme J della Definizione 3 sia un insieme convesso limitato, si adotta invece la definizione di *discrepanza isotropica*.

Definizione 5. Sia C un insieme convesso limitato. Sia $X = \{x_1, \dots, x_N\} \subset C$ una sequenza finita di punti di C . Si definisce **discrepanza isotropica** dell'insieme di punti X la quantità

$$J(X) = \sup_{B \in \mathcal{C}} \left| \frac{\#(B, X)}{N} - \lambda_d(B) \right|, \quad (1.18)$$

dove \mathcal{C} è la famiglia di tutti i sottoinsiemi convessi di C .

Valgono le seguenti relazioni [44].

$$D(X) \leq J(X) \leq 4dD(X)^{1/d}.$$

Il seguente teorema è un risultato classico della teoria della distribuzione uniforme delle sequenze, e ci permette di chiarire la relazione intercorrente tra le varie nozioni di discrepanza e quella di distribuzione uniforme di una sequenza [44].

Teorema 4. Sia $J = \prod_{i=1}^d [\alpha_i, \beta_i) = \{x \in \mathbb{R}^d : \alpha_i \leq x_i < \beta_i\}$. Data una sequenza $X \subset J$, le seguenti proprietà sono equivalenti:

- (i) X è uniformemente distribuito in J ;

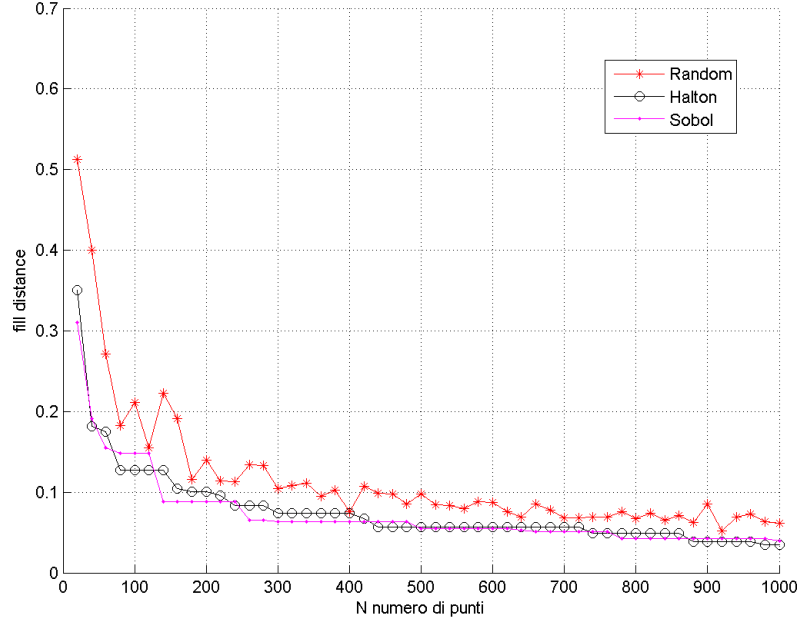


Figura 1.3: Valori della fill distance dei punti random, di Halton e di Sobol su $[0, 1]^2$, al variare del numero di punti N .

$$(ii) \lim_{N \rightarrow \infty} D(X) = 0.$$

$$(iii) \lim_{N \rightarrow \infty} D^*(X) = 0$$

$$(iv) \lim_{N \rightarrow \infty} J(X) = 0$$

Veniamo finalmente alla definizione di *sequenza quasi-random* o a *bassa discrepanza*.

Definizione 6. Una sequenza $X = \{x_1, \dots, x_N\}$ si dice **quasi-random** o a **bassa discrepanza** se

$$D_N(X) \leq c(\log N)^k N^{-1}, \quad (1.19)$$

dove c e k sono costanti indipendenti da N , ma che possono dipendere dalla dimensione d .

Esistono diverse sequenze a bassa discrepanza, come quella di *Halton* e quella di *Sobol* [48, 44, 9], di cui nelle prossime sezioni andremo a descrivere la costruzione. Nella figura 1.2 è possibile vedere il confronto tra le discrepanze stellate dei punti random, di Halton e di Sobol nel quadrato unitario al variare del numero di punti. Le due sequenze di punti quasi-random risultano avere discrepanza (stellata) inferiore a quella dei punti random, come ci si aspettava.

Oltre alla discrepanza, esiste un'altro elemento che misura il grado di distribuzione uniforme dei punti, la *fill distance* o *meshsize* [36].

Definizione 7. Dato un insieme $X = \{x_1, \dots, x_N\}$ di punti su un dominio compatto Ω , si definisce **fill distance** o **meshsize** la quantità

$$h_{X,\Omega} := \sup_{x \in \Omega} \min_{x_i \in X} \|x - x_i\|_2.$$

La fill distance indica quanto bene i punti riempiono in maniera omogenea il dominio Ω , e da un punto di vista geometrico può essere vista come il raggio della più grande palla vuota che può essere collocata tra i punti di X . La figura 1.3 riporta i valori delle fill distance dei punti random, di Halton e di Sobol nel quadrato unitario al variare del numero di punti. Anche in questo caso, i punti di Halton e di Sobol tendono ad avere fill distance minori di quelle dei punti random.

1.2.3 Stime d'errore

Se $\Omega = [0, 1]^d$, l'errore assoluto di un metodo di integrazione quasi-Monte Carlo è

$$\epsilon(f) := \left| \frac{1}{N} \sum_{i=1}^N f(x_i) - \int_{[0,1]^d} f(x) dx \right|. \quad (1.20)$$

La disuguaglianza di Koksma-Hlawka è il risultato più importante sull'errore di integrazione del metodo quasi-Monte Carlo, e fa uso del concetto di *funzione a variazione limitata* [48].

Definizione 8. Sia $f : [0, 1]^d \rightarrow \mathbb{R}$ una funzione e J un sottointervallo di $[0, 1]^d$. Sia $\Delta(f; J)$ una somma alternata dei valori di f nei vertici di J (ovvero i valori della funzione f in vertici adiacenti hanno segno opposto). La **variazione di f su $[0, 1]^d$ in senso di Vitali** è definita da

$$V^{(d)}(f) = \sup_{\mathcal{P}} \sum_{J \in \mathcal{P}} |\Delta(f; J)|, \quad (1.21)$$

dove l'estremo superiore è esteso a tutte le partizioni \mathcal{P} di $[0, 1]^d$ in sottointervalli. Nel caso in cui la derivata parziale $\frac{\partial^d f}{\partial t_1 \dots \partial t_d}$ sia continua su $[0, 1]^d$, la (1.21) è equivalente a

$$V^{(d)}(f) = \int_0^1 \dots \int_0^1 \left| \frac{\partial^d f}{\partial t_1 \dots \partial t_d} \right| dt_1 \dots dt_d.$$

Per $1 \leq k \leq d$, $1 \leq i_1 < i_2 < \dots < i_k \leq d$, sia $V^{(k)}(f; i_1, \dots, i_k)$ la variazione in senso di Vitali della restrizione di f alla faccia k -dimensionale $\{(u_1, \dots, u_d) \in [0, 1]^d : u_j = 1 \text{ per } j \neq i_1, \dots, i_k\}$. Allora

$$V(f) = \sum_{k=1}^d \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq d} V^{(k)}(f; i_1, \dots, i_k)$$

è detta **variazione di f su $[0, 1]^d$ in senso di Hardy-Krause**. Se $V(f)$ è finita, f si dice a **variazione limitata in senso di Hardy-Krause**.

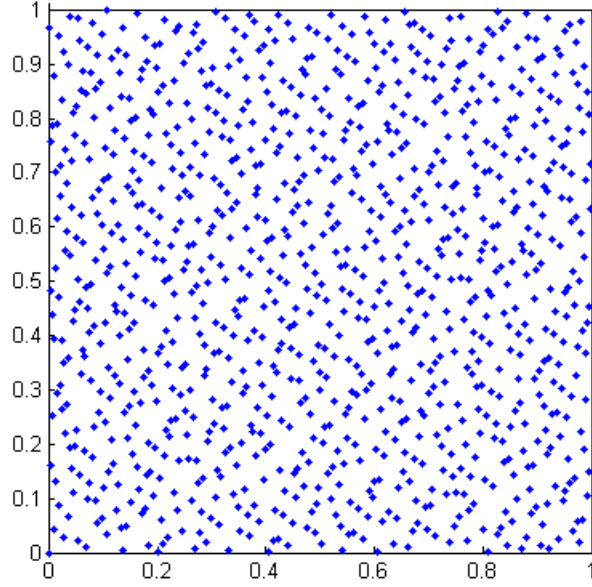


Figura 1.4: Punti di Halton su $[-1, 1]^2$, $N = 1000$.

Teorema 5. (Koksma-Hlawka). *Sia $f : [0, 1]^d \rightarrow \mathbb{R}$ una funzione a variazione limitata nel senso di Hardy-Krause, e $X = \{x_1, \dots, x_N\} \subset [0, 1]^d$ una sequenza qualsiasi. Allora*

$$\epsilon(f) \leq V(f) D_N^*(X) \quad (1.22)$$

dove $D_N^*(X)$ è la discrepanza stellata di X .

Mettendo insieme il Teorema 5 con la Definizione 6 e la relazione (1.17), si deduce il seguente corollario.

Corollario 6. *L'errore di integrazione per i metodi quasi-Monte Carlo è dell'ordine $\mathcal{O}((\log N)^k N^{-1})$, con k dipendente dalla dimensione d .*

I metodi quasi-Monte Carlo, dunque, potenzialmente forniscono un risultato migliore nei problemi di integrazione, perché danno un errore che è minore di quello del metodo Monte Carlo, e di cui conosciamo un upper bound anziché una stima probabilistica. Tuttavia, il fatto che l'esponente k dipenda dalla dimensione d , fa sì che per dimensioni molto grandi i metodi quasi-Monte Carlo possano perdere in accuratezza, dando risultati molto simili a quelli del metodo Monte Carlo. Lo stesso vale per funzioni f poco regolari, a causa del fattore $V(f)$ che è tanto più buono quanto più regolare è la funzione integranda. Esistono tecniche finalizzate al superamento di questi due problemi, per approfondimenti si veda [9].

1.2.4 Sequenza di Halton

La sequenza di Halton d -dimensionale è una sequenza a bassa discrepanza che si ottiene dalla sequenza di Van Der Corput 1-dimensionale.

Si consideri un intero $p \geq 2$. Ogni intero $n \geq 0$ può essere univocamente riscritto in base p

$$n = \sum_{i=0}^{\infty} a_i(n) p^i, \quad (1.23)$$

dove $a_i(n) \in \{0, \dots, p-1\}$ per ogni $i \geq 0$, ed esiste un $k \geq 0$ tale che $a_i(n) = 0$ per $i > k$, ovvero la somma (1.23) è finita.

Per ogni $n \geq 0$, costruiamo la trasformazione $\phi_p : \mathbb{N} \rightarrow [0, 1]$ definita da

$$\phi_p(n) = \sum_{i=0}^{\infty} a_i(n) p^{-(i+1)}, \quad (1.24)$$

dove $a_i(n), i \geq 0$ sono i coefficienti della scrittura (1.23) di n in base p .

Definizione 9. Sia $p \in \mathbb{N}$ un intero. Si definisce **sequenza di Van Der Corput in base p** la sequenza $h_{p,N}$ di $n+1$ punti su $[0, 1]$

$$h_{p,N} := \{\phi_p(n), n = 0, \dots, N\}. \quad (1.25)$$

Definizione 10. Siano d numeri interi $p_1, \dots, p_d \geq 2$ non necessariamente distinti, dove d è la dimensione dello spazio. Si definisce **sequenza di Halton in base $\mathbf{p}_1, \dots, \mathbf{p}_d$** la sequenza $H_{d,N}$ di $N+1$ punti su $[0, 1]^d$

$$H_{d,N} := \{(\phi_{p_1}(n), \dots, \phi_{p_d}(n)), n = 0, \dots, N\} \quad (1.26)$$

Si può dimostrare che se la sequenza di Halton ha base p_1, \dots, p_d formata da numeri primi tra loro, allora ha discrepanza stellata $D^*(H_{d,N})$ dell'ordine $\mathcal{O}(N^{-1}(\log N)^d)$ [48], e di conseguenza per la (1.17) tale sequenza è pure a bassa discrepanza. Si osservi che i punti di Halton hanno la notevole proprietà di essere annidati, ovvero $H_{d,M} \subset H_{d,N}$ per ogni $M < N$.

1.2.5 Sequenza di Sobol

La *sequenza di Sobol* è un'altra sequenza a bassa discrepanza [8, 42, 55, 1], e fa uso dei polinomi primitivi.

Una sequenza di Sobol $S_{d,N}$ è formata da N punti sul cubo d -dimensionale $[0, 1]^d$. Per generare l' i -esimo punto della sequenza occorre scegliere un polinomio primitivo di grado arbitrario s_i a coefficienti nel campo \mathbb{Z}_2 , ovvero un polinomio della forma

$$x_{s_i} + a_{1,i} x_{s_i-1} + a_{2,i} x_{s_i-2} + \dots + a_{s_i-1,i} x + 1,$$

dove i coefficienti $a_{1,i}, a_{2,i}, \dots, a_{s_i-1,i}$ sono 0 o 1. Si scelgono, inoltre, s_i numeri arbitrari $m_{k,i}$, $1 \leq k \leq s_i$, purché dispari e inferiori a 2^k .

Il polinomio primitivo viene utilizzato per definire la sequenza di interi positivi $\{m_{k,i}\}_{k \geq 1}$ determinati dalla relazione ricorsiva

$$m_{k,i} = 2a_{1,i}m_{k-1,i} \oplus 2^2a_{2,i}m_{k-2,i} \oplus \dots \oplus 2^{s_i-1}a_{s_i-1,i}m_{k-s_i+1,i} \oplus 2^{s_i}m_{k-s_i,i} \oplus m_{k-s_i,i} \quad (1.27)$$

per ogni $k > s_i$. Il simbolo \oplus è l'operatore di somma bit-a-bit modulo 2 (*bitwise exclusive or*), ovvero l'operazione che, dati due numeri, ne considera le scritture

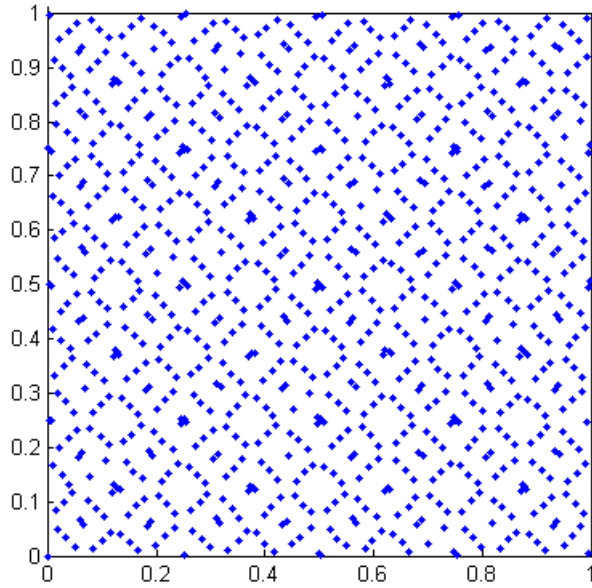


Figura 1.5: Punti di Sobol su $[-1, 1]^2$, $N = 1000$.

in binario e va a sommare a due a due le cifre che si trovano nella medesima posizione, senza tener conto di eventuali riporti. Un esempio di somma bit-a-bit modulo 2 è il seguente

$$44 \oplus 53 = (101100)_2 \oplus (110101)_2 = (011001)_2 = 25,$$

dove $(\cdot)_2$ sta ad indicare la rappresentazione binaria.

Una volta ricavata la sequenza $\{m_{k,i}\}_{k \geq 1}$, si ricavano i cosiddetti *direction numbers*¹ $\{\nu_{k,i}\}_{k \in \mathbb{N}}$ dalla relazione

$$v_{k,i} = \frac{m_{k,i}}{2^k}.$$

La j -esima componente dell' i -esimo punto della sequenza di Sobol è data da

$$x_{i,j} := i_1 \nu_{1,j} \oplus i_2 \nu_{2,j} \oplus \dots, \quad (1.28)$$

dove i_k è la k -esima cifra da destra nella rappresentazione binaria di $i = (\dots i_3 i_2 i_1)_2$.

Esiste una implementazione più efficiente dell'algoritmo originale di Sobol appena descritto, il cosiddetto *Gray code*, ideato da Antonov e Saleev [1].

Definizione 11. *Il Gray code di un intero i è definito come*

$$\text{gray}(i) := i \oplus \left\lfloor \frac{i}{2} \right\rfloor = (\dots i_3 i_2 i_1)_2 \oplus (\dots i_4 i_3 i_2)_2.$$

¹Talvolta, con leggero abuso di linguaggio, vengono chiamati *direction numbers* i numeri $m_{k,i}$, $1 \leq k \leq s_i$.

i	$\text{gray}(i)$
0 = (0000) ₂	(0000) ₂ = 0
1 = (0001) ₂	(0001) ₂ = 1
2 = (0010) ₂	(0011) ₂ = 3
3 = (0011) ₂	(0010) ₂ = 2
4 = (0100) ₂	(0110) ₂ = 6
5 = (0101) ₂	(0111) ₂ = 7
6 = (0110) ₂	(0101) ₂ = 5
7 = (0111) ₂	(0100) ₂ = 4
8 = (1000) ₂	(1100) ₂ = 12
9 = (1001) ₂	(1101) ₂ = 13
10 = (1010) ₂	(1111) ₂ = 15
11 = (1011) ₂	(1110) ₂ = 14
12 = (1100) ₂	(1010) ₂ = 10
13 = (1101) ₂	(1011) ₂ = 11
14 = (1110) ₂	(1001) ₂ = 9
15 = (1111) ₂	(1000) ₂ = 8

Figura 1.6: Calcolo del Gray code dei numeri da 0 a 15.

Il Gray code ha la proprietà che le rappresentazioni binarie di $\text{gray}(i)$ e $\text{gray}(i - 1)$ differiscono soltanto dell'indice della prima cifra 0 che compare da destra nella rappresentazione binaria di $i - 1$. Inoltre, applicare il Gray code agli interi aventi lo stesso numero di cifre nella loro rappresentazione binaria, equivale a farne il riordinamento, come si può vedere nella Figura 1.6.

I punti di Sobol $S_{d,N}$ possono quindi essere generati usando

$$x_{i,j} := g_{i,1}\nu_{1,j} \oplus g_{i,2}\nu_{2,j} \oplus \cdots, \quad (1.29)$$

dove $g_{i,k}$ è la k -esima cifra da destra del Gray code di i in rappresentazione binaria, ovvero $\text{gray}(i) = (\dots g_{i,3}g_{i,2}g_{i,1})_2$.

Equivalentemente, poiché $\text{gray}(i)$ e $\text{gray}(i - 1)$ differiscono di una sola posizione, i punti di Sobol possono essere generati ricorsivamente da

$$x_{0,j} := 0, \quad x_{i,j} := x_{i-1,j} \oplus \nu_{c_{i-1},j}, \quad (1.30)$$

dove c_i è l'indice della prima cifra 0 da destra nella rappresentazione binaria di $i = (\dots i_3i_2i_1)_2$.

Con le due implementazioni (1.29) e (1.30) del metodo Gray code si generano esattamente gli stessi punti dati dalla formula (1.28), ma con un diverso ordine. Tra le due formulazioni del metodo Gray code, quella (1.30) è la più efficiente, grazie alla sua natura ricorsiva.

È stato dimostrato che, scegliendo d differenti polinomi primitivi per calcolare d diversi insiemi di direction numbers, l'insieme dei punti di Sobol $S_{d,N}$ ha discrepanza dell'ordine $\mathcal{O}((\log N)^d)$ [8, 55]. Inoltre, proprio come i punti di Halton, anche i punti di Sobol hanno la notevole proprietà di essere annidati, ovvero $S_{d,M} \subset S_{d,N}$ per ogni $M < N$, e questo è visibile nella formula (1.30).

Capitolo 2

Metodi di cubatura

In questo capitolo introduciamo alcune formule di quadratura su certi domini multivariati aventi grado prefissato di precisione algebrico. In particolare ci soffermeremo sui domini che andremo ad usare negli esperimenti numerici al Capitolo 5, ossia il poligono e la lente, e ne illustreremo le pertinenti routines di Matlab `Polygauss` e `gqlens`, disponibili open-source. Verrà infine fatta una rapida rassegna dei vari metodi di cubatura presenti in letteratura.

2.1 Polygauss

`Polygauss` è una routine implementata in Matlab [61] per eseguire la cubatura sui poligoni, sia convessi sia non convessi.

L'algoritmo di `Polygauss` implementa le idee descritte in [62], fornendo una formula di cubatura del tipo

$$\iint_{\Omega} f(x, y) dx dy \approx \sum_{(\xi, \eta) \in \Xi_{2n-1}} w_{\xi, \eta} f(\xi, \eta), \quad \Omega \subset \mathbb{R}^2 \text{ poligono},$$

esatta per tutti i polinomi bivariati di grado al più $2n - 1$ e stabile, ossia $\sum_{(\xi, \eta) \in \Xi_{2n-1}} |w_{\xi, \eta}|$ è limitato.

L'idea di base è quella di combinare la quadratura univariata di Gauss-Legendre con la formula integrale di Green

$$\iint_{\Omega} f(x, y) dx dy = \oint_{\partial\Omega} \mathcal{F}(x, y) dy, \quad \mathcal{F}(x, y) = \int f(x, y) dx, \quad (2.1)$$

che ha il vantaggio di trasformare un problema 2-dimensionale in uno 1-dimensionale, qualora sia $f \in \mathcal{C}(\Omega)$ e inoltre Ω abbia il bordo regolare a tratti e descritto in senso antiorario.

Ciò che si va a fare è calcolare una primitiva $\mathcal{F}(x, y)$ nella variabile x per la formula (2.1), tramite la quadratura di Gauss-Legendre; dopodiché, grazie alla formula di Green, $\mathcal{F}(x, \cdot)$ viene integrata lungo i lati del poligono Ω utilizzando la quadratura di Gauss-Legendre una seconda volta. Quanto detto è formalizzato nel seguente teorema, di cui omettiamo la dimostrazione (si veda [62]).

Teorema 7. Sia $\Omega \subset \mathbb{R}^2$ la chiusura di un poligono limitato e semplicemente connesso con bordo descritto in senso antiorario dalla sequenza di vertici

$$V_i = (\alpha_i, \beta_i), \quad i = 1, \dots, L, \quad L \geq 3, \\ \partial\Omega = [V_1, V_2] \cup \dots \cup [V_L, V_{L+1}], \quad V_{L+1} = V_1.$$

Siano $f \in \mathcal{C}(\mathcal{R})$ e $\alpha \in \mathbb{R}$ tali che

$$\Omega \subseteq \mathcal{R} = [a, b] \times [c, d], \\ \alpha \in [a, b]. \quad (2.2)$$

Siano inoltre $\{\tau_j^s\}_{1 \leq j \leq s}$ e $\{\lambda_j^s\}_{1 \leq j \leq s}$ rispettivamente i nodi e i pesi della formula di quadratura di Gauss-Legendre con grado di esattezza $2s - 1$ su $[-1, 1]$.

Allora, la seguente formula di cubatura è esatta su Ω per ogni polinomio bivariato di grado al più $2n - 1$

$$I_{2n-1}(f) = \sum_{i \in \mathcal{I}_{\Omega, \alpha}} \sum_{j=1}^{n_i} \sum_{k=1}^n w_{ijk} f(\xi_{ijk}, \eta_{ij}), \quad (2.3)$$

$$\mathcal{I}_{\Omega, \alpha} = \{i : \Delta\beta_i \neq 0\} \cap \{i : \alpha_i \neq \alpha \text{ oppure } \alpha_{i+1} \neq \alpha\} \subseteq \{1, \dots, L\},$$

$$n_i = \begin{cases} n, & \Delta\alpha_i = 0 \\ n+1, & \Delta\alpha_i \neq 0 \end{cases}$$

ovvero $\mathcal{I}_{\Omega, \alpha}$ è l'insieme degli indici dei lati che non sono ortogonali alla linea $x = \alpha$ e che non giacciono su di essa. I nodi e i pesi sono dati da

$$\xi_{ijk} = \frac{x_i(\tau_j^{n_i}) - \alpha}{2} \tau_k^n + \frac{x_i(\tau_j^{n_i}) + \alpha}{2}, \quad x_i(t) = \frac{\Delta\alpha_i}{2} t + \frac{\alpha_i + \alpha_{i+1}}{2}, \\ \eta_{ij} = y_i(\tau_j^{n_i}), \quad y_i(t) = \frac{\Delta\beta_i}{2} t + \frac{\beta_i + \beta_{i+1}}{2}, \\ w_{ijk} = \frac{\Delta\beta_i}{4} (x_i(\tau_j^{n_i}) - \alpha) \lambda_j^{n_i} \lambda_k^n,$$

dove con Δ indichiamo l'operatore differenza in avanti.

Ponendo $m = \text{card}(\mathcal{I}_{\Omega, \alpha})$, si ha che il numero totale dei nodi di cubatura è

$$\mathcal{V} = \mathcal{V}_{n, \Omega, \alpha} = n \sum_{i \in \mathcal{I}_{\Omega, \alpha}} n_i$$

con

$$\frac{L}{2} n^2 \leq mn^2 \leq \mathcal{V} \leq mn(n+1) \leq Ln(n+1). \quad (2.4)$$

Inoltre, valgono la seguente stima della stabilità

$$\sum_{i \in \mathcal{I}_{\Omega, \alpha}} \sum_{j=1}^{n_i} \sum_{k=1}^n |w_{ijk}| \leq C_{\Omega, \alpha} = \max_{i \in \mathcal{I}_{\Omega, \alpha}} |\alpha - \alpha_i| \left(\sum_{i \in \mathcal{I}_{\Omega, \alpha}} |\Delta\beta_i| \right)$$

e la seguente stima dell'errore

$$\left| \iint_{\Omega} f(x, y) dx dy - I_{2n-1}(f) \right| \leq (\text{meas}(\Omega) + C_{\Omega, \alpha}) E_{2n-1}(f; \mathcal{R}) \\ E_{2n-1}(f; \mathcal{R}) = \min_{p \in \mathbb{P}_{2n-1}^2} \|f - p\|_{\infty, \mathcal{R}}. \quad (2.5)$$

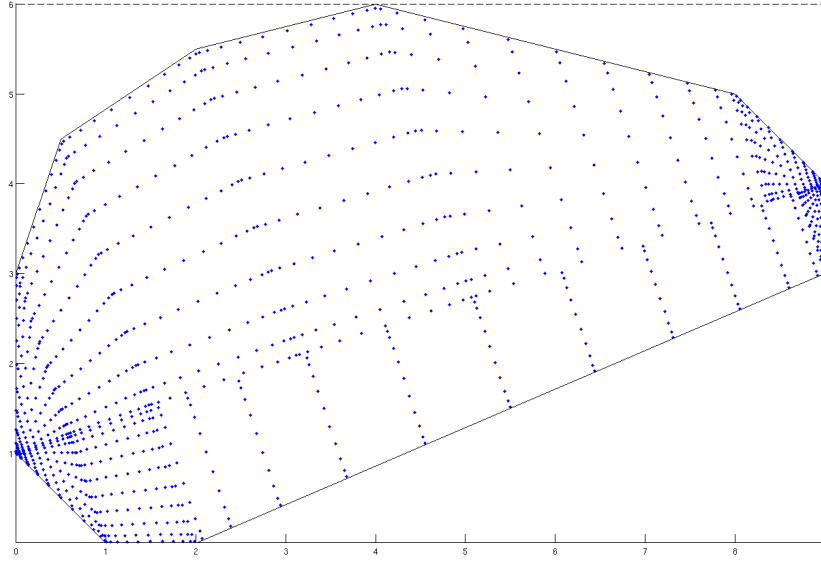


Figura 2.1: Esempio di distribuzione dei nodi in un poligono convesso con 10 lati ed $n = 10$. I nodi sono 1100 e hanno tutti pesi positivi.

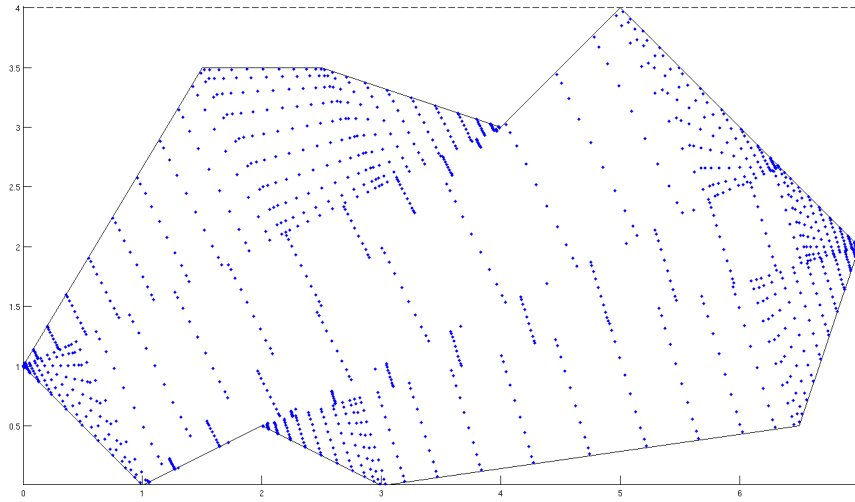


Figura 2.2: Esempio di distribuzione dei nodi in un poligono non convesso con 10 lati ed $n = 10$. I nodi sono 1430, di cui 1320 hanno pesi positivi (circa il 92%). Il rapporto $\sum |w_i| / \sum w_i$ è pari a 1.

La formula di cubatura (2.3) è esatta per ogni polinomio bivariato di grado al più $2n-1$, indipendentemente dalla base polinomiale scelta. Inoltre, se applicata a un polinomio di grado r , scegliendo

$$n \geq \frac{r+1}{2}$$

si ottiene l'integrale (in aritmetica esatta) richiesto.

Un altro fatto apprezzabile del Teorema 7 è che, grazie alle disuguaglianze (2.4), fornisce un lower bound e un upper bound per il numero dei nodi. Ad esempio, per avere una formula di cubatura di grado $n = 10$ su un poligono con 4 lati, e quindi esatta per i polinomi di grado al più 19, sarà necessario un numero di nodi $200 \leq V \leq 440$, da paragonarsi con la dimensione dello spazio polinomiale $\eta_n = \dim(\mathbb{P}_n^2) = \binom{n+2}{2} = 55$. Se invece prendiamo un poligono con 10 lati, il numero di nodi V sale notevolmente, essendo compreso tra 500 e 1100. Dalla (2.4), quindi, si vede bene come la formula di cubatura (2.3) dipenda strettamente dal numero di lati del poligono; d'altra parte, maggiore è il numero di lati di un poligono, maggiormente complessa può essere la sua struttura geometrica.

Un particolare da tenere presente, è che in generale i nodi di cubatura possono non essere all'interno del poligono Ω , bensì solo all'interno del bounding box \mathcal{R} (cioè il più piccolo rettangolo con lati paralleli agli assi e contenente il poligono). È per questo motivo che si suppone che f sia continua su \mathcal{R} , e che la stima dell'errore (2.5) fa uso dell'approssimazione polinomiale uniforme su tutto \mathcal{R} . Tuttavia, esiste una grande classe di poligoni per cui si può ovviare a tale problema operando un semplice cambio di coordinate. I poligoni in questione devono avere la proprietà di contenere una “base-line” ℓ la cui intersezione con il poligono, se esiste, sia connessa, e tale che ciascuna retta h ortogonale ad essa abbia intersezione connessa con il poligono, contenente il punto $\ell \cap q$. Appartengono a questa classe tutti i poligoni convessi (come “base-line” ℓ , ad esempio, si può prendere la retta su cui giace la diagonale maggiore), ma anche molti poligoni non convessi. Il cambio di variabili nell'integrazione consiste in una rotazione che porta ℓ ad essere parallela al nuovo asse delle y . Il parametro α che si andrà a scegliere in (2.2) sarà l'ascissa in cui la “base-line” ℓ ruotata interseca il nuovo asse x .

Da un punto di vista geometrico, una volta individuata la baseline, si rende possibile operare una scomposizione del poligono in sottopoligoni, effettuare la cubatura su ciascuno di questi, e sommarne infine tutti i contributi con l'accortezza di introdurre (eventuali) sottrazioni nei casi di sovrapposizione. Combinando la formula di Green con la discretizzazione gaussiana di un'opportuna primitiva, questo procedimento viene eseguito “in automatico” nella formula di cubatura (2.3), il che la rende particolarmente pratica.

L'algoritmo **Polygauss** originario permetteva di dare in input una coppia di punti che definissero la “base-line”, qualora l'utente fosse in grado di fornirne una, altrimenti sceglieva di default una delle due distanze massime tra i vertici del poligono. Tuttavia, potevano verificarsi casi di poligoni non convessi i cui nodi ricadevano in parte fuori dal poligono. La versione aggiornata di tale algoritmo (**polygauss_2014**, disponibile in [61]), invece, opera una partizione del poligono in sottoregioni convesse, e, per ognuna delle sottoregioni, ricerca una “base-line” tramite la quale ricavare nodi e pesi. In questo modo, tutti i nodi del poligono

stanno necessariamente al suo interno, e la formula di cubatura (2.3) si ottiene sommando tutti i contributi delle varie sottoregioni.

Le figure 2.1 e 2.2 mostrano alcuni esempi numerici di quanto detto: si sono scelti 2 poligoni con 10 lati, di cui uno convesso e uno non convesso (che peraltro soddisfa alla proprietà della “base-line”). Il primo ha pesi solo positivi, mentre il poligono non convesso ne ha anche di negativi. I nodi di cubatura prodotti, inoltre, tendono ad addensarsi lungo le “base-line”, lungo i lati e in prossimità dei vertici.

2.2 gqlens

La routine Matlab `gqlens` [28] calcola nodi e pesi per la formula di cubatura per domini di tipo lente piana, ovvero l’intersezione di due dischi parzialmente sovrapposti, con raggi arbitrari, di lunghezze non necessariamente uguali.

L’algoritmo di `gqlens` è stato sviluppato dapprima in [27] e si basa sulla formula gaussiana trigonometrica “subperiodica” studiata precedentemente in [29]. Si tratta di una formula di quadratura con $n + 1$ nodi (angoli) e pesi positivi, esatta sullo spazio $(2n + 1)$ -dimensionale dei polinomi trigonometrici di grado al più n

$$\mathbb{T}_n([- \omega, \omega]) = \text{span}\{1, \cos(k\theta), \sin(k\theta), \ 1 \leq k \leq n, \ \theta \in [- \omega, \omega]\},$$

dove $0 < \omega \leq \pi$. La riportiamo brevemente nel seguente risultato [29].

Teorema 8. *Siano $\{(\xi_j, \lambda_j)\}_{1 \leq j \leq n+1}$ le coppie di nodi e pesi positivi della formula di quadratura gaussiana algebrica per la funzione peso*

$$w(x) = \frac{2 \sin(\omega/2)}{\sqrt{1 - \sin^2(\omega/2)x^2}}, \quad x \in (-1, 1).$$

Allora vale la formula

$$\int_{-\omega}^{\omega} f(\theta) d\theta = \sum_{j=1}^{n+1} \lambda_j f(\theta_j), \quad \forall f \in \mathbb{T}_n([- \omega, \omega]), \ 0 < \omega \leq \pi, \quad (2.6)$$

dove

$$\theta_j = 2 \arcsin(\sin(\omega/2)\xi_j) \in (-\omega, \omega), \ j = 1, 2, \dots, n+1. \quad (2.7)$$

Si noti che nel caso in cui si abbia $\omega = \pi$, si ottiene la già nota formula di Gauss-Chebyshev, e gli angoli $\{\theta_j\}_{1 \leq j \leq n+1}$ sono i consueti $n+1$ angoli equispaziati in $(-\pi, \pi)$ con i pesi di Gauss-Chebyshev. La formula (2.6) deve il nome di “subperiodica” proprio al fatto che è possibile scegliere valori ω minori di π , restringendo quindi la quadratura all’intervallo angolare $[-\omega, \omega]$, che in generale è strettamente contenuto nel periodo $[-\pi, \pi]$.

La formula (2.6) è stata utilizzata in [29] per costruire formule gaussiane prodotto con esattezza polinomiale per settori circolari, segmenti circolari, zone circolari e lenti simmetriche sul disco. Successivamente, in [27] è stata applicata anche per domini di tipo lenti piane asimmetriche e doppie bolle, ovvero rispettivamente intersezione ed unione di dischi con raggio possibilmente diverso.

Nel caso delle lenti asimmetriche, che è il caso che ci interesserà nel Capitolo 5, la formula di cubatura nasce da quella per segmenti circolari, in quanto la

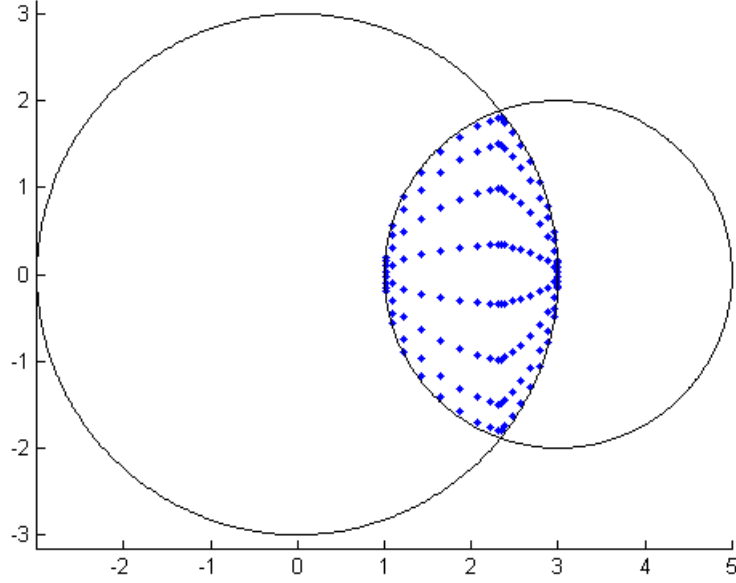


Figura 2.3: Esempio di distribuzione dei nodi in una lente piana asimmetrica, con $n = 10$ e 144 nodi, tutti a pesi positivi.

lente può essere vista come l'unione di due segmenti circolari con la corda in comune. Tenendo presente che, a meno di traslazioni e rotazioni, il segmento circolare può essere parametrizzato da

$$S = \{(x, y) = (R \cos(\theta), Rt \sin(\theta)), \theta \in [0, \omega], t \in [-1, 1]\}, \quad (2.8)$$

dove $0 < \omega \leq \pi$, vale il seguente risultato per la formula gaussiana prodotto per il segmento circolare [27].

Teorema 9. *Per ogni segmento circolare S parametrizzato come in (2.8), vale la seguente formula prodotto*

$$\iint_S f(x, y) dx dy = \sum_{j=1}^{\lfloor \frac{n+2}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n+1}{2} \rfloor} W_{ij} f(x_{ij}, y_{ij}), \quad \forall f \in \mathbb{P}_n^2, \quad (2.9)$$

con \mathbb{P}_n^2 lo spazio dei polinomi bivariati di grado totale al più n , e

$$W_{ij} = R^2 \sin^2(\theta_j) w_j^{GL} \lambda_j, \quad (x_{ij}, y_{ij}) = (R \cos(\theta_j), Rt_i^{GL} \sin(\theta_j)), \quad (2.10)$$

dove $\{(\theta_j, \lambda_j)\}$ sono gli angoli e i pesi della formula gaussiana trigonometrica (2.6) con grado di esattezza $n+2$ su $[-\omega, \omega]$, e $\{(t_i^{GL}, w_i^{GL})\}$ sono i nodi e i pesi della formula di Gauss-Legendre con grado di esattezza n su $[-1, 1]$.

La formula (2.9)-(2.10) usa un numero di nodi che varia tra $(n+2)(n+2)/4$ e $(n+1)(n+3)/4$. Il numero dei nodi è quindi dell'ordine $n^2/4$, ovvero circa la metà rispetto alla dimensione dello spazio polinomiale \mathbb{P}_n^2 , che è $(n+1)(n+2)/2 \sim n^2/2$. Ciò significa che se si vuole fare la cubatura, ad esempio, su un segmento circolare con grado $n = 20$, sono necessari 121 nodi, mentre la dimensione dello

spazio polinomiale è $\eta_n = \dim(\mathbb{P}_n^2) = \binom{n+2}{2} = 231$. D'altra parte, essendo la lente (ma anche le doppie bolle) unione di due segmenti circolari, il numero di nodi per la formula di cubatura con grado di esattezza n è doppio rispetto a quello per il segmento circolare, ovvero compreso tra $(n+2)(n+2)/2$ e $(n+1)(n+3)/2$, e ha ordine $n^2/2$.

Si noti, inoltre, che prendendo $\omega = \pi$ nella formula (2.9)-(2.10), si ottiene una formula alternativa per la cubatura sull'intero disco, con $n(n+1)$ nodi e grado di esattezza $2n-1$.

Per la cubatura su segmenti circolari, è stata sviluppata la funzione Matlab `gqcircsegm` [30], che calcola nodi e pesi (2.10) tramite la formula (2.6), la quale a sua volta è implementata dalla function `trigauss` [26]. La routine Matlab `gqlens` [28], invece, calcola nodi e pesi della formula di cubatura per l'intersezione di due dischi arbitrari, tramite la (2.9)-(2.10), effettuando le opportune traslazioni e rotazioni. L'implementazione ricorre a `trigauss` per calcolare gli angoli e i pesi gaussiani trigonometrici, e lo fa separatamente per ciascun arco circolare, ma non usa direttamente `gqcircsegm` per determinare i nodi e i pesi di Gauss-Legendre. Inoltre, nel caso singolare in cui l'intersezione abbia misura zero (ossia i dischi non si sovrappongono), `gqlens` restituisce un solo nodo e con peso nullo. Per i domini di tipo doppia bolla, similmente, è stata implementata in Matlab la function `gqdbubble` [28], che lavora in maniera analoga a `gqlens`.

Riguardo alla velocità di convergenza della formula (2.9)-(2.10), vale la seguente stima dell'errore per ogni fissato $k > 0$ e ogni funzione integranda f sufficientemente regolare

$$\iint_Q f(x, y) dx dy = \sum_{j=1}^{N_n} \sum_{i=1}^{\lceil \frac{n+1}{2} \rceil} \sigma_{ij} f(a_{ij}, b_{ij}) + \mathcal{O}(n^{-k}),$$

dove Q è qualsiasi dominio di tipo segmento circolare (con $N_n = \lceil (n+2)/2 \rceil$) oppure lente piana (con $N_n = 2\lceil (n+2)/2 \rceil$) oppure doppia bolla (con $N_n = 2\lceil (n+2)/2 \rceil$), e $\{(a_{ij}, b_{ij})\}$ e $\{\sigma_{ij}\}$ sono rispettivamente i nodi e i pesi della formula di cubatura. Questa stima è resa possibile dal fatto che i pesi sono tutti positivi e che sia il segmento circolare sia la lente piana sia la doppia bolla sono compatti di Jackson.

Definizione 12. Sia $Q \subset \mathbb{R}^d$ un sottoinsieme tale che $Q = \overline{\text{int}Q}$. Q è detto **compatto di Jackson** se ammette una disuguaglianza di Jackson, ovvero per ogni $k \in \mathbb{N}$ esistono un intero positivo m_k ed una costante positiva c_k tali che

$$n^k \text{dist}_Q(f, \mathbb{P}_n^d) \leq c_k \sum_{|i| \leq m_k} \|D^i f\|_Q, \quad n > k, \quad \forall f \in \mathcal{C}^{m_k}(Q)$$

dove $\text{dist}_Q(f, \mathbb{P}_n^d) = \inf\{\|f - p\|_Q, p \in \mathbb{P}_n^d\}$.

Nella figura 2.3 si può vedere la distribuzione dei nodi di cubatura calcolati da `gqlens` per una lente piana asimmetrica con grado $n = 15$.

2.3 Altri metodi di cubatura

Negli ultimi anni sono stati sviluppati molteplici metodi e routines su Matlab per la cubatura numerica. In Matlab esistono le function `dblquad` [65] e `integral2` [66], che calcolano integrali doppi su domini rettangolari.

Recentemente L. F. Shampine ha sviluppato `TwoD` [52, 53], una routine di Matlab che calcola integrali su regioni piane in modo più efficiente e su una maggiore varietà di domini rispetto a `dblquad`.

S. E. Mousavi e altri [47] hanno realizzato un algoritmo numerico che calcola in modo efficiente formule di quadratura per l'integrazione di polinomi bivariati su poligoni arbitrari, ricorrendo a nozioni di teoria dei gruppi e di ottimizzazione numerica.

M. G. Trimbilas e R. T. Trimbilas [69] hanno recentemente implementato un algoritmo adattativo per la cubatura su triangoli.

R. Cools assieme a collaboratori [17, 23, 18, 20, 19, 22] ha studiato formule di cubatura su diversi domini, tra cui il quadrato, il disco, il triangolo e il cubo n -dimensionale. Inoltre, in [14, 15, 21] ha proseguito il lavoro iniziato da Stroud nella cubatura [64], sviluppando formule di quadrature sul cubo, sulla sfera e sul semplice n -dimensionali, e sugli spazi n -dimensionali individuati dalle funzioni peso $\exp(-x_1^2 - x_2^2 - \dots - x_n^2)$ ed $\exp(-\sqrt{-x_1^2 - x_2^2 - \dots - x_n^2})$. Queste formule sono raccolte nel sito della *Encyclopaedia of Cubature Formulas* [16].

Di particolare interesse per le applicazioni, sono i risultati ottenuti da I. H. Sloan e R. Womersley con vari collaboratori relativamente a formule di cubatura sulla sfera [54, 41] e metodi di tipo Quasi-Montecarlo.

Il gruppo di ricerca CAA di Padova-Verona sull'Approssimazione Costruttiva e Applicazioni ha sviluppato numerose routines per la cubatura multivariata. Oltre alla function `Polygauss` [62, 61] per la quadratura sui poligoni, sono state implementate formule di cubatura minimali sul quadrato [37]. Per il rettangolo, esistono il package `Padua2DM` [13, 12], che esegue l'interpolazione e la cubatura sui punti di Padova, e il package `XuPad2D` [10, 11], un'interfaccia Matlab che permette l'iperinterpolazione sui punti di Xu. Per il disco, è stata sviluppata la routine `GreenDisk` [50, 49], che esegue la cubatura senza mesh su dischi e su corone circolari a partire da dati sparsi tramite le Thin-Plate Splines. Il CAA ha inoltre implementato su Matlab la function `Splinegauss` [51, 57], una formula di cubatura di Gauss-Green su domini 2-dimensionali arbitrari con bordo regolare a tratti, ovvero su poligoni curvilinei il cui bordo può essere tracciato con delle spline. Esiste poi la routine `ChebfunGauss` [51, 57], una function di Matlab che usa una versione adattata di `Splinegauss` per domini bivariati definiti analiticamente ed approssimati dal package `Chebfun`. Per domini di tipo rettangoli geografici, ovvero regioni di \mathbb{S}^d definite da latitudine e longitudine, è disponibile la routine di Matlab `GeorectQ` [40, 39], che calcola formule di quadratura di tipo algebrico.

Recentemente, inoltre, il CAA ha implementato la quadratura gaussiana prodotto per una grande varietà di sezioni del disco e di regioni generate da dischi parzialmente sovrapposti e archi ellittici [29, 27, 24]: `gqcircsect` [30] per i settori circolari, `gqcircsegm` [30] per i segmenti circolari, `gqcirczone` [30] per le zone circolari, `gqsymmlens` [30] per le lenti simmetriche, `gqdbubble` [28] per le doppie bolle, ossia l'unione di due dischi parzialmente sovrapposti, `gqlens` [28] per le lenti asimmetriche, ovvero l'intersezione di due dischi parzialmente sovrapposti, `gqlune` [28] per le lune, vale a dire la differenza di due dischi parzialmente sovrapposti, e `gqellblend` [25] per regioni generate dalla combinazione di archi ellittici. Queste routine si basano tutte sulla function `trigauss` [26], che determina angoli e pesi della formula di cubatura gaussiana su intervalli di lunghezza minore di 2π , con grado di esattezza n nello spazio $(2n + 1)$ -dimensionale dei polinomi trigonometrici di grado al più n . Accanto a `trigauss`, un'altra func-

tion che il CAA ha sviluppato per la quadratura trigonometrica subperiodica è la routine `trigquad` [6, 7], che calcola nodi e pesi (positivi) per formule di cubatura algebriche.

Per la cubatura sul cubo, alternativamente alle formule di Gauss-Chebyshev prodotto tensoriale, alla Clenshaw-Curtis tensoriale e alla Gauss-legendre tensoriale, il CAA ha implementato il pacchetto `Hyper3` [35, 32], che raccoglie diverse formule nuove di cubatura: una formula algebrica di grado $2n + 1$ per la misura di Chebyshev prodotto sul cubo, e una di tipo Clenshaw-Curtis non tensoriale. Per piramidi, coni e solidi di rotazione esiste il package `3dWAM` [33, 34]. Per il toro e la sfera, inoltre, sono stati studiati anche metodi di cubatura adattativa [46]. In [31], infine, sono state sviluppate formule di cubatura sul cilindro che fanno uso dei punti approssimati di Fekete.

Capitolo 3

La function `lsqnonneg`

Nel capitolo precedente sono stati presentati diversi metodi che calcolano nodi e pesi di cubatura. In alcuni dei domini considerati, le formule proposte, pur avendo una cardinalità maggiore od uguale alla dimensione dello spazio polinomiale, avevano pesi positivi e grado algebrico di precisione prefissato. In generale ciò non è sempre disponibile. Si potrebbero avere formule algebriche con alcuni pesi negativi. Ricordiamo che tali formule risultano in generale “meno stabili” di quelle a pesi positivi, motivo per cui l’ideale sarebbe avere un metodo che determini solo nodi di cubatura che abbiano i corrispondenti pesi positivi.

Il metodo `lsqnonneg` è una funzione implementata in Matlab [67] che risolve problemi ai minimi quadrati non negativi. Dandole in ingresso un’opportuna matrice e un opportuno vettore, quindi, `lsqnonneg` può essere utilizzata per estrarre il minimo numero di nodi X con pesi solo positivi da un insieme di punti Y sufficientemente grande, e fare su questi la cubatura a meno di un errore ϵ rispetto alla cubatura sull’intero insieme Y .

In questo capitolo si vuole illustrare la teoria su cui è basato il metodo `lsqnonneg`, ovvero l’algoritmo NNLS. Verranno inoltre forniti alcuni dettagli implementativi della routine.

3.1 L’algoritmo NNLS

La funzione Matlab `lsqnonneg` rientra nella più ampia classe di solutori per *problemi NNLS* (*Nonnegative Least Squares*), ovvero problemi di minimi quadrati non negativi. Si tratta di particolari *problemi LSI* (*Least Squares Inequality*), ovvero di minimi quadrati con vincoli lineari di disuguaglianza, e in questo capitolo andremo a descriverne brevemente la teoria relativa [45].

Definizione 13. Siano A una matrice $m \times n$, $b \in \mathbb{R}^m$ un vettore, G una matrice $r \times n$, e $h \in \mathbb{R}^r$ un vettore. Si definisce **problema LSI** (*Least Squares Inequality*), ovvero **problema di minimi quadrati con vincoli lineari di disuguaglianza**, il seguente problema di ottimizzazione:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \|Ax - b\| \\ & Gx \geq h. \end{aligned} \tag{3.1}$$

Ad esempio, prendendo

- $A = V^T$, dove V^T è la trasposta della matrice di Vandermonde V calcolata su un insieme di punti X di cardinalità n , in una data base $\{p_j\}_{1 \leq j \leq m}$;
- $b = \mathbf{m}$, dove $\mathbf{m} = (m_1, \dots, m_m)^T$ è il vettore dei momenti $m_j = \int_{\Omega} p_j(x) d\mu(x)$, $1 \leq j \leq m$, nella base $\{p_j\}_{1 \leq j \leq m}$;
- $G = I$, dove I è la matrice identità;
- $h = (0, \dots, 0)^T$

si ottiene il seguente problema LSI

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \|V^T x - \mathbf{m}\| \\ x_i \geq 0 \quad i = 1, \dots, n, \end{aligned} \quad (3.2)$$

che è il caso di nostro interesse. Il vettore x soluzione di (3.2), infatti, è il vettore dei pesi non negativi associati ai nodi X . In generale, i pesi strettamente positivi saranno in numero $\hat{n} \leq n$, e quindi andranno a determinare altrettanti nodi $\{\xi_1, \dots, \xi_{\hat{n}}\} \subseteq X$, ottenendo un numero possibilmente minore di nodi rispetto a quello iniziale, e tutti con pesi positivi.

Vediamo ora la definizione di problema NNLS.

Definizione 14. *Siano A una matrice $m \times n$, e $b \in \mathbb{R}^m$. Si definisce **problema NNLS (Nonnegative Least Squares)**, ovvero **problema di minimi quadrati non negativi**, il seguente problema di ottimizzazione:*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \|Ax - b\| \\ x \geq 0. \end{aligned} \quad (3.3)$$

Il problema (3.2) è evidentemente un problema NNLS, con $A = V^T$ e $b = \mathbf{m}$, dove V è la matrice di Vandermonde e \mathbf{m} il vettore dei momenti.

Il teorema di Kuhn-Tucker dà la caratterizzazione delle soluzioni dei problemi LSI (3.1).

Teorema 10. (Kuhn-Tucker). *Un vettore $\hat{x} \in \mathbb{R}^n$ è soluzione del problema LSI (3.1) se e solo se esistono un vettore $\hat{y} \in \mathbb{R}^r$ e una partizione degli interi $\mathcal{I} = \{1, \dots, r\}$ in due sottoinsiemi \mathcal{Z} ed \mathcal{P} , tali che*

$$G^T \hat{y} = A^T (A \hat{x} - b) \quad (3.4)$$

$$G_i \hat{x} = h_i \text{ per } i \in \mathcal{Z}, \quad G_i \hat{x} > h_i \text{ per } i \in \mathcal{P} \quad (3.5)$$

$$\hat{y}_i \geq 0 \text{ per } i \in \mathcal{Z}, \quad \hat{y}_i = 0 \text{ per } i \in \mathcal{P} \quad (3.6)$$

dove G_i è la riga i -esima della matrice G , h_i è la componente i -esima del vettore h , e \hat{y}_i è la componente i -esima del vettore \hat{y} . Il vettore \hat{y} è detto soluzione duale del problema LSI (3.1), mentre \hat{x} è la soluzione primale.

Essendo il problema NNLS un tipo di problema LSI, per la risoluzione di (3.3) si applica il Teorema 10, da cui si ottiene il seguente corollario.

Corollario 11. *Un vettore $\hat{x} \in \mathbb{R}^n$ è soluzione del problema NNLS (3.3) se e solo se esistono un vettore $\hat{y} \in \mathbb{R}^r$ e una partizione degli interi $\mathcal{I} = \{1, \dots, r\}$ in due sottoinsiemi \mathcal{Z} ed \mathcal{P} , tali che*

$$\hat{y} = A^T(A\hat{x} - b) \quad (3.7)$$

$$\hat{x}_i = 0 \text{ per } i \in \mathcal{Z}, \quad \hat{x}_i > 0 \text{ per } i \in \mathcal{P} \quad (3.8)$$

$$\hat{y}_i \geq 0 \text{ per } i \in \mathcal{Z}, \quad \hat{y}_i = 0 \text{ per } i \in \mathcal{P} \quad (3.9)$$

dove \hat{x}_i è la componente i -esima del vettore \hat{x} , e \hat{y}_i è la componente i -esima del vettore \hat{y} . Il vettore \hat{y} è detto soluzione duale del problema NNLS (3.3), mentre \hat{x} è la soluzione primale.

L'algoritmo NNLS risolve i problemi NNLS (3.3) utilizzando il corollario precedente. Infatti, presi in ingresso una matrice A di dimensioni $m \times n$ e un vettore $b \in \mathbb{R}^m$, ricerca una coppia di vettori $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$ tali che soddisfino alle condizioni (3.7), (3.8), (3.9). In realtà, il vettore y determinato dall'algoritmo NNLS è l'opposto del vettore \hat{y} del Corollario 11, perché è definito da

$$y := A^T(b - Ax),$$

essendo calcolato dal residuo $r = b - Ax$. Nell'algoritmo NNLS, quindi, la condizione (3.9) viene sostituita dalla richiesta

$$\hat{y}_i \leq 0 \text{ per } i \in \mathcal{Z}, \quad \hat{y}_i = 0 \text{ per } i \in \mathcal{P}. \quad (3.10)$$

I due insiemi di indici \mathcal{P} e \mathcal{Z} vengono inizializzati rispettivamente come

$$\mathcal{P} := \emptyset, \quad \mathcal{Z} := \{1, 2, \dots, n\}.$$

Le variabili x_i indicizzate in \mathcal{Z} vengono mantenute con valore zero, mentre quelle indicizzate in \mathcal{P} sono libere di assumere valori diversi da zero. In particolare, se una variabile x_i ha valore negativo, l'algoritmo provvede a modificarne il valore, portandola a un valore positivo (tramite un opportuno passo α) oppure ponendola uguale a zero, e in quest'ultimo caso il suo indice viene rimosso dall'insieme \mathcal{P} e spostato in \mathcal{Z} . Ad ogni iterazione dell'algoritmo, viene utilizzato un vettore ausiliare $z \in \mathbb{R}^n$ che memorizza di volta in volta la candidata soluzione primale. Al termine dell'esecuzione, l'algoritmo dà in output la soluzione primale x e la soluzione duale y .

ALGORITMO NNLS(A, b)

1. Inizializza $\mathcal{P} := \emptyset$, $\mathcal{Z} := \{1, 2, \dots, n\}$, $x := 0$.
2. Calcola il vettore $y := A^T(b - Ax)$.
3. Se $\mathcal{Z} := \emptyset$ oppure $y_i \leq 0 \forall i \in \mathcal{Z}$, vai al punto 12.
4. Trova un indice $j \in \mathcal{Z}$ tale che $y_j = \max_{i \in \mathcal{Z}} y_i$.
5. Sposta l'indice j dall'insieme \mathcal{Z} all'insieme \mathcal{P} .
6. Sia $A_{\mathcal{P}}$ la matrice $m \times n$ definita da

$$\text{colonna } i\text{-esima di } A_{\mathcal{P}} := \begin{cases} \text{colonna } i\text{-esima di } A & \text{se } i \in \mathcal{P} \\ 0 & \text{se } i \in \mathcal{Z} \end{cases}$$

Calcola il vettore z come soluzione del problema ai minimi quadrati $A_{\mathcal{P}}z \approx b$. (Si noti che soltanto le componenti z_i con $i \in \mathcal{P}$ sono determinate da questo problema).

Poni $z_i := 0 \ \forall i \in \mathcal{Z}$.

7. Se $z_i > 0 \ \forall i \in \mathcal{P}$, poni $x := z$ e vai al punto 2.

8. Trova un indice $k \in \mathcal{P}$ tale che

$$\frac{x_k}{x_k - z_k} = \min \left\{ \frac{x_i}{x_i - z_i} : z_i \leq 0, i \in \mathcal{P} \right\}.$$

9. Poni $\alpha := x_k / (x_k - z_k)$.

10. Poni $x := x + \alpha(z - x)$.

11. Sposta dall'insieme \mathcal{P} all'insieme \mathcal{Z} tutti gli indici $i \in \mathcal{P}$ tali che $x_i = 0$. Vai al punto 6.

12. Fine.

Al termine dell'algoritmo, il vettore x soddisfa

$$x_i > 0, \quad i \in \mathcal{P} \quad (3.11)$$

$$x_i = 0, \quad i \in \mathcal{Z}. \quad (3.12)$$

D'altra parte, il vettore y soddisfa

$$y_i = 0 \quad i \in \mathcal{P} \quad (3.13)$$

$$y_i \leq 0 \quad i \in \mathcal{Z} \quad (3.14)$$

e vale

$$y = A^T(b - Ax). \quad (3.15)$$

Inoltre, dalle equazioni (3.12), (3.13), (3.15) discende che x è soluzione del problema ai minimi quadrati

$$A_{\mathcal{P}}x \approx b. \quad (3.16)$$

Le relazioni (3.11), (3.12), (3.13), (3.14), (3.15) costituiscono le condizioni (3.7), (3.8), (3.9), e pertanto garantiscono che i vettori calcolati x e y sono proprio rispettivamente la soluzione primale e la soluzione duale del problema NNLS (3.3).

3.2 Convergenza dell'algoritmo NNLS

Nella sezione precedente abbiamo illustrato l'algoritmo NNLS su cui si basa il metodo `lsqnonneg` di Matlab. Tuttavia, un requisito fondamentale di ogni algoritmo è che questo converga in un numero finito di passi.

Partiamo con il seguente lemma tecnico [45].

Lemma 12. *Siano A una matrice $m \times n$ di rango n , e $b \in \mathbb{R}^m$ un vettore tale che*

$$A^T b = \begin{bmatrix} 0_{n-1} \\ \omega \end{bmatrix} \quad (3.17)$$

dove 0_{n-1} è il vettore nullo $(0, \dots, 0)^T$ di dimensione $n-1$, e $\omega \in \mathbb{R}$, $\omega > 0$.
Se \hat{x} è la soluzione ai minimi quadrati di $Ax \approx b$, allora

$$\hat{x}_n > 0$$

dove \hat{x}_n è la n -esima componente di \hat{x} .

Dimostrazione del lemma: Sia Q una matrice $m \times m$ ortogonale, tale che azzeri il blocco subdiagonale delle prime $n-1$ colonne di A , ovvero

$$Q \cdot A = \begin{bmatrix} R & s \\ O & t \end{bmatrix}, \quad Q \cdot b = \begin{bmatrix} u \\ v \end{bmatrix} \quad (3.18)$$

dove R è una matrice $(n-1) \times (n-1)$ triangolare superiore e non singolare, O è una matrice $(n-1) \times (m-n+1)$ nulla, e $s, u \in \mathbb{R}^{n-1}$ e $t, v \in \mathbb{R}^{m-n+1}$ sono vettori colonna. L'esistenza di una tale matrice Q è assicurata dal teorema della fattorizzazione QR (Theorem 3.11, p. 11, [45]).

Poiché Q è ortogonale, si ha

$$A^T b = A^T Q^T Q b = (QA)^T Q b = \begin{bmatrix} R^T & O^T \\ s^T & t^T \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} R^T u \\ s^T u + t^T v \end{bmatrix} \quad (3.19)$$

La condizione (3.17) assieme alla (3.19) implica

$$R^T u = 0_{n-1} \quad (3.20)$$

$$s^T u + t^T v = \omega > 0. \quad (3.21)$$

Ora, poiché R è non singolare, dalla (3.20) discende che

$$u = 0 \quad (3.22)$$

e quindi la (3.21) si riduce a

$$t^T v = \omega > 0.$$

Essendo Q matrice ortogonale, x è soluzione di $Ax \approx b$ se e solo se è soluzione di $QAx \approx Qb$; inoltre, poiché per (3.18) e (3.22)

$$QAx \approx Qb \iff \begin{bmatrix} R & s \\ O & t \end{bmatrix} x \approx \begin{bmatrix} 0_{n-1} \\ v \end{bmatrix},$$

segue che la n -esima componente \hat{x}_n del vettore soluzione \hat{x} è la soluzione ai minimi quadrati del problema ridotto

$$tx_n \approx v. \quad (3.23)$$

Poiché la pseudoinversa del vettore colonna t è $t^T/(t^T t)$, la soluzione del problema (3.23) è

$$\hat{x}_n = \frac{t^T v}{t^T t} = \frac{\omega}{t^T t} > 0$$

e completa la dimostrazione del lemma. \square

Possiamo ora enunciare il seguente teorema [45].

Teorema 13. *L'algoritmo NNLS ha convergenza finita.*

Traccia della dimostrazione: Si osservi innanzitutto che l'algoritmo NNLS può essere visto come la composizione di un loop principale, che chiamiamo Loop A, e di un loop interno, che chiamiamo invece Loop B.

Il Loop B comprende i passi 6 – 11, ha una singola entrata al passo 6 e una singola uscita al passo 7. Il Loop A, invece, è composto dai passi 2 – 5 e dal Loop B; inoltre, ha un'entrata al passo 2 e un'uscita al passo 3.

Al passo 2 del Loop A, l'insieme degli indici \mathcal{P} identifica le componenti del vettore x che sono positive, mentre le componenti di x indicizzate in \mathcal{Z} sono nulle.

Nel Loop A, l'indice j selezionato al passo 4 sceglie un coefficiente non presente nell'insieme \mathcal{P} ma che diventa positivo (per il Lemma 12) se introdotto nella soluzione. Questo coefficiente viene portato nel vettore z al passo 6 del Loop B, dove z è un vettore ausiliario che a ogni iterazione del Loop B memorizza la nuova candidata soluzione. Se tutte le altre componenti di z indicizzate in \mathcal{P} sono rimaste positive, allora al passo 7 l'algoritmo pone $x := z$ e ritorna all'inizio del Loop A, ovvero al passo 2. Durante questo processo l'insieme \mathcal{P} è stato incrementato, mentre l'insieme \mathcal{Z} è stato diminuito per il passaggio dell'indice j da \mathcal{Z} a \mathcal{P} .

A questo punto, una volta tornati al passo 2, possono verificarsi due casi:

1. la sequenza di istruzioni dei passi 2 – 7 si ripete, con la somma di un ulteriore coefficiente positivo j ad ogni iterazione del Loop A, finché il test di terminazione al passo 3 viene soddisfatto e si esce dall'algoritmo.
2. nel passo 6 qualche coefficiente indicizzato in \mathcal{P} è diventato nullo o negativo nel vettore z , e di conseguenza il passo 7 obbliga l'algoritmo a procedere per l'intero Loop B. Al passo 8 viene quindi scelto un opportuno passo $0 < \alpha \leq 1$ lungo cui muovere la soluzione x , che diventa $x + \alpha(z - x)$. Il passo α viene calcolato in modo tale che sia il più grande possibile e che mantenga la nuova soluzione x non negativa. Il Loop B viene ripetuto finché viene soddisfatta la condizione di terminazione al passo 7.

Fatte tutte queste considerazioni preliminari, la convergenza finita del Loop B può essere dimostrata provando che:

- tutte le operazioni all'interno del Loop B sono ben definite;
- esiste almeno un indice (quello che nell'algoritmo abbiamo chiamato k) che viene rimosso dall'insieme \mathcal{P} ogniqualvolta il passo 11 viene eseguito;
- z_j è sempre positiva (per il Lemma 12).

In tal modo, l'uscita dal Loop B al passo 7 è garantita dopo al massimo $\tau - 1$ iterazioni del Loop B, dove τ indica il numero di indici presenti nell'insieme \mathcal{P} quando il Loop B è entrato.

Per provare la convergenza finita del Loop A, invece, è sufficiente mostrare che la funzione norma del residuo

$$\rho(x) = \|b - Ax\|$$

assume valore strettamente minore ogni volta che il passo 2 viene raggiunto. In tal modo, infatti, al passo 2 il vettore x e il suo corrispondente insieme $\mathcal{P} = \{i : x_i > 0\}$ sono differenti da tutte le istanze precedenti di x e \mathcal{P} al passo

2, e poiché \mathcal{P} è un sottoinsieme di $\{1, 2, \dots, n\}$ ed esiste solo un numero finito di tali sottoinsiemi, il Loop A deve necessariamente terminare dopo un numero finito di iterazioni. \square

3.3 L'implementazione

La routine `lsqnonneg` [67] ha un'implementazione semplice, in quanto traduce fedelmente l'Algoritmo NNLS nell'equivalente codice Matlab.

La function `lsqnonneg`, infatti, si compone di due cicli `while` innestati: il primo corrisponde a quello che abbiamo chiamato Loop A nella dimostrazione del Teorema 13, e comprende i passi 2–5 dell'Algoritmo NNLS; il ciclo interno, invece, è quello che abbiamo definito Loop B e include i passi 6–11.

L'indice $j \in \mathcal{Z}$ al Passo 4 viene calcolato tramite il metodo `max` e nel caso in cui il massimo sia raggiunto da più indici diversi, viene scelto il primo; in maniera analoga, l'indice $k \in \mathcal{P}$ del passo 8 viene determinato con il metodo `min`. Il passo 6 che determina la candidata soluzione z è implementato tramite l'operatore “backslash” di Matlab.

La function `lsqnonneg` prende in ingresso la matrice A e il vettore b del sistema lineare di cui si vuole calcolare il vettore soluzione positivo x . Inoltre, prevede che si possano dare in input un numero massimo di iterazioni per il Loop B, e una tolleranza $tol \approx 0$ da usare per le condizioni di uscita dal Loop A ($y_i \leq 0 \forall i \in \mathcal{Z}$) e dal Loop B ($z_i \geq 0 \forall i \in \mathcal{P}$). Vengono restituiti in output il vettore soluzione x , il residuo, la norma di quest'ultimo, il numero di iterazioni compiute e la variabile duale y .

Capitolo 4

I punti approssimati di Fekete (AFP)

Nel Capitolo 1 si è mostrato come i metodi di integrazione quasi-Monte Carlo garantiscano risultati migliori rispetto all'integrazione Monte Carlo, sia per l'entità minore dell'errore di cubatura, sia per la sua natura deterministica che permette di avere un upper bound dell'errore anziché una stima probabilistica. Tuttavia, esattamente come il metodo Monte Carlo, i metodi di integrazione quasi-Monte Carlo presentano lo svantaggio di dover maneggiare una grande mole di campioni.

D'altro canto, i metodi per la cubatura visti al Capitolo 2, pur lavorando con un numero molto piccolo di campioni, hanno il limite di dipendere fortemente dalla geometria del dominio su cui sono implementati. Esistono, infatti, domini su cui non sono ancora stati ideati analoghi algoritmi per la cubatura, come ad esempio l'unione arbitraria finita di più regioni diverse.

Questo è il motivo per cui si è deciso di tentare un nuovo approccio, che consiste nel combinare l'integrazione quasi-Monte Carlo con il concetto di *punti approssimati di Fekete*. I punti di Fekete sono quei punti che massimizzano, in valore assoluto, il determinante di Vandermonde di un dato dominio compatto, e per costruzione risultano essere buoni per l'interpolazione e per la quadratura in qualunque dominio. Tuttavia, al momento sono noti analiticamente solo in pochi casi; inoltre, determinarli numericamente è molto difficile nonché costoso, poiché si tratta di risolvere problemi di ottimizzazione multivariata non lineare.

Per ovviare a questo problema, sono stati sviluppati i cosiddetti punti approssimati di Fekete (AFP). Si tratta di punti estratti da mesh di punti che discretizzano il dominio, e che vengono calcolati tramite la semplice fattorizzazione QR con pivoting per colonne. Inoltre, nel caso di *Weakly Admissible Meshes* (WAMs), ovvero griglie di punti quasi-ottimali per l'approssimazione ai minimi quadrati, si è visto che gli AFP hanno distribuzione asintoticamente uguale ai punti di Fekete rispetto alla misura d'equilibrio pluripotenziiale-teoretica del dominio compatto. Tutto questo fa sì che gli AFP costituiscano un valido insieme di punti per l'interpolazione e la quadratura, e ha quindi motivato in questa direzione la nostra ricerca di una nuova tecnica per la cubatura.

In questo capitolo verranno presentati i punti approssimati di Fekete e il metodo di cubatura tramite AFP. Verrà inoltre fatta una breve introduzione

ai punti di Fekete e alle Weakly Admissible Meshes. Per approfondimenti, si vedano [59, 4, 5] e i riferimenti ivi presenti.

4.1 I punti di Fekete

Prima di dare la definizione di punti di Fekete, occorre fornire alcuni richiami.

Sia $\Omega \subset \mathbb{C}^d$ un compatto. Si consideri $\mathbb{P}_n^d(\Omega)$ lo spazio dei polinomi su \mathbb{C}^d di grado minore od uguale ad n ristretto all'insieme Ω , e indichiamo con $\eta_n = \binom{n+d}{d}$ la dimensione dello spazio $\mathbb{P}_n^d(\Omega)$. In questa trattazione supporremo che Ω sia \mathbb{P}_n^d -determinante, ovvero che se $p \in \mathbb{P}_n^d$ è identicamente nullo su Ω , allora è identicamente nullo anche su \mathbb{C}^d .

Dati un insieme ordinato di punti $\mathbf{x} = \{x_1, \dots, x_k\} \subset \mathbb{C}^d$ e un insieme ordinato di polinomi $\mathbf{p} = \{p_1, \dots, p_m\} \subseteq \mathbb{P}_n^d(\mathbb{C}^d)$, ricordiamo la definizione di *matrice di Vandermonde*:

$$V(\mathbf{x}; \mathbf{p}) = V(x_1, \dots, x_k; p_1, \dots, p_m) := [p_j(x_i)]_{1 \leq i \leq k, 1 \leq j \leq m} \in \mathbb{C}^{k \times m}.$$

Nel caso sia $m = k$, la matrice di Vandermonde è quadrata e usiamo la seguente notazione per indicare il suo determinante

$$vdm(\mathbf{x}; \mathbf{p}) = vdm(x_1, \dots, x_k; p_1, \dots, p_m) := \det(V(x_1, \dots, x_k; p_1, \dots, p_m)).$$

Si consideri ora $S_n = \text{span}(q_j)_{1 \leq j \leq \eta_n}$ uno spazio finito-dimensionale di funzioni continue linearmente indipendenti. Tipicamente, si prendono $S_n = \mathbb{P}_n^d(\Omega)$, $\eta_n = \dim(\mathbb{P}_n^d(\Omega))$. Un insieme finito $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_{\eta_n}\} \subseteq \Omega$ si dice *unisolvante* per l'interpolazione su S_n se

$$vdm(\boldsymbol{\xi}, \mathbf{q}) \neq 0$$

e in tal caso le funzioni

$$\ell_j(x) = \frac{vdm(\xi_1, \dots, \xi_{j-1}, x, \xi_{j+1}, \dots, \xi_{\eta_n}; \mathbf{q})}{vdm(\xi_1, \dots, \xi_{j-1}, \xi_j, \xi_{j+1}, \dots, \xi_{\eta_n}; \mathbf{q})}, \quad j = 1, \dots, \eta_n \quad (4.1)$$

costituiscono una base cardinale dello spazio S_n , ovvero $\ell_j(\xi_k) = \delta_{jk}$ e la funzione $L_{S_n} : C(\Omega) \rightarrow S_n$ definita da

$$L_{S_n} f(x) = \sum_{j=1}^{\eta_n} f(\xi_j) \ell_j(x)$$

interpolata ogni funzione $f \in S_n$ nei punti $\{\xi_1, \dots, \xi_{\eta_n}\}$. In termini matriciali, la base cardinale $\boldsymbol{\ell} = \{\ell_1, \dots, \ell_{\eta_n}\}$ si ottiene da quella originale \mathbf{q} mediante il sistema

$$\boldsymbol{\ell}^T = L \mathbf{q}^T,$$

dove $L := ((V(\boldsymbol{\xi}; \mathbf{q}))^{-1})^T$.

Infine, si definisce *costante di Lebesgue* di interpolazione nei punti $\{\xi_1, \dots, \xi_{\eta_n}\}$ la quantità

$$\Lambda_n := \|L_{S_n}\| = \max_{x \in \Omega} \sum_{j=1}^{\eta_n} |\ell_j(x)| = \max_{x \in \Omega} \|L \mathbf{q}^T(x)\|_1. \quad (4.2)$$

Possiamo ora dare la definizione di punti di Fekete.

Definizione 15. Siano $\Omega \subset \mathbb{C}^d$ un compatto e $S_n = \text{span}(q_j)_{1 \leq j \leq \eta_n}$ uno spazio finito-dimensionale di funzioni continue linearmente indipendenti. I punti $\{\xi_1, \dots, \xi_{\eta_n}\} \subset \Omega$ si dicono *punti di Fekete* se sono unisolventi per l'interpolazione su S_n e inoltre massimizzano il valore assoluto del denominatore di (4.1) in Ω^{η_n} .

Sia i punti di Fekete sia la costante di Lebesgue Λ_n non dipendono dalla scelta della base in S_n . Il motivo è che i determinanti delle matrici di Vandermonde variano di un fattore indipendente dai punti, che è il determinante della matrice di trasformazione tra le due basi, per cui nella formula (4.1) compare a numeratore e a denominatore il medesimo fattore, che va a semplificarsi.

I punti di Fekete hanno la proprietà che

$$\|\ell_j\|_\infty \leq 1 \quad j = 1, \dots, \eta_n, \quad (4.3)$$

che, assieme alla (4.3) fornisce il seguente upper bound della costante di Lebesgue per i punti di Fekete

$$\Lambda_n \leq \eta_n.$$

Gli esperimenti numerici hanno tuttavia dimostrato che questo upper bound è molto peggiore dei risultati che si ottengono al calcolatore.

Un'altra proprietà molto importante è che i punti di Fekete sono asintoticamente equidistribuiti rispetto alla misura d'equilibrio pluripotenziiale di Ω [2].

Tuttavia, come si è anticipato prima, i punti di Fekete sono un insieme di punti non semplici da determinare, sia da un punto di vista analitico che numerico. Analiticamente, infatti, sono noti solo in pochissimi casi: nell'intervallo i punti di Fekete sono i punti di Gauss-Lobatto, con costante di Lebesgue $\Lambda_n = \mathcal{O}(\log n)$; nel cerchio complesso sono i punti equispaziati, con $\Lambda_n = \mathcal{O}(\log n)$; nel cubo con l'interpolazione prodotto tensoriale sono i punti di Gauss-Lobatto prodotto tensoriale, con $\Lambda_n = \mathcal{O}(\log^d n)$. Quanto al calcolo numerico dei punti di Fekete, invece, si tratta di risolvere un problema di ottimizzazione non lineare in $\eta_n \times d$ variabili, per cui anche questa strada si mostra impraticabile; i punti di Fekete sono stati determinati numericamente solo in pochi casi, e per un numero limitato di gradi di interpolazione n .

Per ovviare a questi problemi si è tentato l'approccio di calcolare i punti di Fekete su una discretizzazione del dominio, anziché sul dominio vero e proprio, passando così da un problema non lineare sul continuo a uno non lineare combinatorio. Il quesito successivo diventa quindi quale tipo di discretizzazione del dominio sia più conveniente adottare, ed è in questo frangente che entrano in scena le Admissible Meshes.

4.2 Weakly Admissible Meshes (WAMs)

Definizione 16. Sia $\Omega \subset \mathbb{C}^d$ un insieme compatto \mathbb{P}_n^d -determinante. Una successione di sottoinsiemi discreti $\mathcal{A}_n \subset \Omega$ è detta **Weakly Admissible Mesh (WAM)** se soddisfa

$$\|p\|_\Omega \leq C(\mathcal{A}_n) \|p\|_{\mathcal{A}_n} \quad \forall p \in \mathbb{P}_n^d(\Omega), \quad (4.4)$$

dove sia $\text{card}(\mathcal{A}_n) \geq \eta_n$ sia $C(\mathcal{A}_n)$ hanno crescita al più polinomiale con n .

Se $C(\mathcal{A}_n)$ è limitata, allora la sequenza $\{\mathcal{A}_n\}_n$ è detta **Admissible Mesh (AM)**.

Le principali caratteristiche delle WAMs possono essere riassunte nelle seguenti proprietà:

- (i) $C(\mathcal{A}_n)$ è invariante per trasformazioni affini.
- (ii) Ogni successione di insiemi di interpolazione unisolventi la cui costante di Lebesgue abbia crescita al più polinomiale con n è una WAM, e $C(\mathcal{A}_n)$ è la costante di Lebesgue.
- (iii) Ogni successione di superinsiemi di una WAM le cui cardinalità abbiano crescita al più polinomiale con n è una WAM, ed ha la stessa costante $C(\mathcal{A}_n)$.
- (iv) Un'unione finita di WAMs è una WAM per la corrispondente unione dei compatti, e $C(\mathcal{A}_n)$ è il massimo delle corrispettive costanti.
- (v) Un prodotto cartesiano finito di WAMs è una WAM per il corrispondente prodotto dei compatti, e $C(\mathcal{A}_n)$ è il prodotto delle corrispettive costanti.
- (vi) In \mathbb{C}^d una WAM sul bordo $\partial\Omega$ è una WAM su K .
- (vii) Data una funzione polinomiale π_s di grado s , si ha che $\pi_s(\mathcal{A}_{ns})$ è una WAM per $\pi_s(\Omega)$, e ha costante $C(\mathcal{A}_{ns})$.
- (viii) Se Ω soddisfa a una disuguaglianza polinomiale di Markov del tipo $\|\nabla p\|_\Omega \leq Mn^r \|p\|_\Omega$, allora ammette una AM con $\mathcal{O}(n^{rd})$ punti.
- (ix) Approssimazione polinomiale ai minimi quadrati di $f \in \mathcal{C}(\Omega)$: l'interpolante polinomiale ai minimi quadrati $L_{\mathcal{A}_n} f$ su una WAM soddisfa

$$\|f - L_{\mathcal{A}_n} f\| \lesssim C(\mathcal{A}_n) \sqrt{\text{card}(\mathcal{A}_n)} \min\{\|f - p\|_\Omega \mid p \in \mathbb{P}_n^d(\Omega)\}.$$

- (x) Punti di Fekete: la costante di Lebesgue dei punti di Fekete estratti da una WAM è limitata da $\Lambda_n \leq \eta_n C(\mathcal{A}_n)$, che è il bound classico per i punti di Fekete moltiplicato per il fattore $C(\mathcal{A}_n)$. Inoltre, la loro distribuzione asintotica è la medesima dei punti di Fekete sul continuo, nel senso che le corrispondenti misure di probabilità convergono debolmente-* alla misura d'equilibrio pluripotenziale di Ω .

Le proprietà qui elencate costituiscono un valido strumento per costruire AMs e, soprattutto, WAMs su una grande varietà di domini. Per citarne solo qualcuno, nel caso bivariato si possono ottenere AMs sui convessi compatti, e WAMs sul disco, sul triangolo, sui trapezi e su qualsiasi poligono che possa essere suddiviso in triangoli o trapezi tramite algoritmi standard di geometria computazionale [4].

4.3 I punti approssimati di Fekete (AFP)

La proprietà (x) delle WAMs vista nella sezione precedente è quella che autorizza ad usare le WAMs per l'approssimazione ai minimi quadrati delle funzioni continue. D'altra parte, determinare i punti di una WAM è un problema NP-hard, perché si tratta di risolvere un problema di ottimizzazione discreta non lineare.

Per ovviare a questo problema, si è trovato il metodo euristico che andremo ora a presentare, e che, con l'ausilio di semplici strumenti di algebra lineare come la fattorizzazione QR, riesce a dare una buona soluzione approssimata in tempi ragionevolmente brevi [59, 4]. I punti calcolati tramite questo algoritmo sono i cosiddetti *punti approssimati di Fekete* (*Approximate Fekete Points, AFP*).

Dato un insieme compatto $\Omega \subset \mathbb{C}^d$ tale che sia \mathbb{P}_n^d -determinante, si consideri una WAM $\{\mathcal{A}_n\}_n$ di Ω

$$\mathcal{A}_n = \mathbf{a} = \{a_1, \dots, a_M\},$$

con $M \geq \eta_n = \dim(\mathbb{P}_n^d)$. Siano $\mathbf{p} = \{p_j\}_{1 \leq j \leq \eta_n}$ una base di $\mathbb{P}_n^d(\Omega)$ e $V(\mathbf{a}; \mathbf{p}) := [p_j(a_i)], 1 \leq i \leq M, 1 \leq j \leq \eta_n$ la matrice di Vandermonde associata ai punti $\{a_1, \dots, a_M\}$.

Richiamiamo, inoltre, la definizione di *volume di una matrice*: data la matrice A in generale rettangolare, il suo volume è definito dalla quantità $\text{vol}(A) = \sqrt{\det(A^T A)}$.

Riportiamo qui di seguito in pseudo-codice l'algoritmo greedy per il calcolo degli AFP [4].

ALGORITMO AFP GREEDY

1. $V_0 = V(\mathbf{a}; \mathbf{p}); T_0 = I$; (inizializzazione)
 for $k = 0, \dots, s - 1$ (ortogonalizzazione di $V(\mathbf{a}; \mathbf{p})$)
 - $V_k = Q_k R_k; P_k = \text{inv}(R_k)$;
 - $V_{k+1} = V_k P_k; T_{k+1} = T_k P_k$;
 end;
 2. $\text{ind} = []$; $W = V_s^T$; (inizializzazione)
 for $k = 1, \dots, \eta_n$ (algoritmo greedy per selezionare la submatrice $\eta_n \times \eta_n$ di volume massimo)
 a) “seleziona la colonna con norma massima $\text{col}_{i_k}(W)$ ”;
 $\text{ind} = [\text{ind}, i_k]$;
 b) “rimuovi da ogni colonna di W la sua proiezione ortogonale su col_{i_k} ”;
 end;
 3. $\xi = \mathbf{a}(\text{ind})$; (estrazione degli AFP dalla WAM)

Il punto 1 dell'algoritmo opera un numero $s \geq 0$ di ortogonalizzazioni successive della matrice di Vandermonde $V(\mathbf{a}; \mathbf{p})$, per correggerne il malcondizionamento; è noto, infatti, che le matrici di Vandermonde possano avere un numero di condizionamento tipicamente molto elevato, specialmente quando usano basi non ortogonali. L'esperienza mostra che in genere sono sufficienti una o due

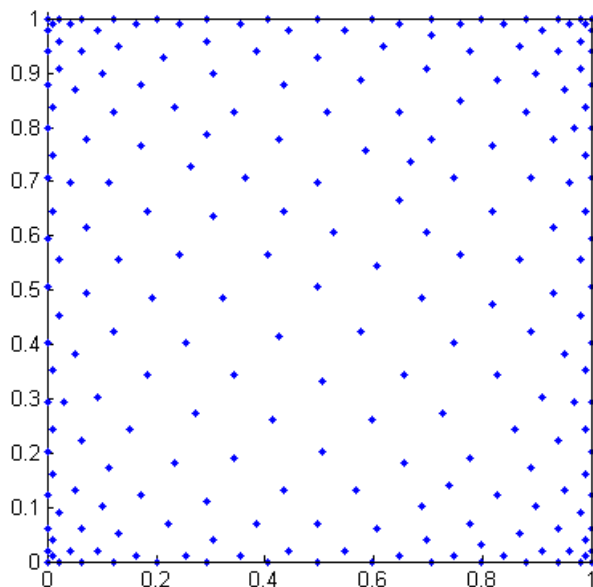


Figura 4.1: Punti approssimati di Fekete sul quadrato $[0, 1]^2$, con grado $n = 20$.

ortogonalizzazioni successive per portare il malcondizionamento a un livello accettabile, a meno che la matrice $V(\mathbf{a}; \mathbf{p})$ sia eccessivamente mal condizionata. Quello che si va a fare è un cambiamento di base: tramite la matrice di trasformazione T_s , si passa dalla base originaria $\mathbf{p} = \{p_1, \dots, p_{\eta_n}\}$ alla base discreta $\phi = (\phi_1, \dots, \phi_{\eta_n}) = \mathbf{p}T_s$ ortonormale rispetto al prodotto interno $\langle f, g \rangle = \sum_{i=1}^{\eta_n} f(a_i)g(a_i)$. In realtà non è necessario che la nuova base ϕ sia ortonormale, quanto piuttosto che la rispettiva matrice di Vandermonde sia sufficientemente ben condizionata.

Il punto 2 è il vero cuore dell'algoritmo ed è qui che viene utilizzata la fattorizzazione QR. In Matlab, infatti, l'algoritmo greedy del punto 2 è implementato tramite le istruzioni

$$\mathbf{w} = W \setminus \mathbf{b}; \quad \text{ind} = \text{find}(\mathbf{w} \neq \mathbf{0}); \quad (4.5)$$

dove \mathbf{b} è un qualunque vettore in \mathbb{R}^{η_n} con componenti *tutte non nulle*; tipicamente si prende $\mathbf{b} = (1, \dots, 1)^T$, ma, come si è appena detto, tale scelta è arbitraria. La fattorizzazione QR con pivoting per colonne entra in gioco quando viene eseguito l'operatore “backslash”, che in Matlab fa uso proprio di questo metodo per risolvere i sistemi sottodeterminati. Poiché le M righe della matrice di Vandermonde V corrispondono agli M punti della mesh \mathcal{A}_n , mentre le η_n colonne di V corrispondono agli η_n elementi della base \mathbf{p} , il punto 2a dell'algoritmo greedy dice che estrarre i punti approssimati di Fekete $\xi = \mathbf{a}(\text{ind})$ equivale a selezionare le η_n righe di V tali che il determinante della rispettiva sottomatrice $\eta_n \times \eta_n$ sia massimo in valore assoluto. Questo fatto può essere letto come il voler ricercare l'insieme delle η_n righe di V il più possibile ortogonali tra loro.

Dalla formula (4.5) risulta evidente come gli AFP dipendano strettamente dalla base di polinomi scelta (al contrario dei veri punti di Fekete). Non solo, la bontà dell'approssimazione con cui vengono calcolati dipenderà necessariamente

dal condizionamento di V . È per questo motivo che il punto 1 è indispensabile: permette, infatti, di passare da una base poco buona a una più adatta, ovvero ortogonale (o quasi), e quindi con rispettiva matrice di Vandermonde ben condizionata. Tuttavia, se l'insieme di partenza $\mathcal{A}_n \subset \Omega$ da cui vengono estratti gli AFP è una WAM, allora i punti approssimati di Fekete su \mathcal{A}_n hanno la medesima distribuzione asintotica dei punti di Fekete rispetto alla misura d'equilibrio pluripotenziale-teoretica di Ω , qualunque sia la base polinomiale con cui vengono calcolati gli AFP [3].

4.4 Cubatura sugli AFP

Siano $\Omega \subset \mathbb{R}^d$ un insieme compatto e $\mathbb{P}_n^d = \text{span}(p_j)_{1 \leq j \leq \eta_n}$ il sottospazio dei polinomi d -variati con grado minore od uguale ad n ristretti a Ω , dove $\eta_n = \dim(\mathbb{P}_n^d)$. Dato un insieme di punti $\mathbf{x} = \{x_1, \dots, x_M\} \subset \Omega$ con $M \geq \eta_n$, costruiamo la formula di cubatura algebrica con grado di esattezza n per ogni data misura μ su Ω [59]. A tal fine occorre risolvere il sistema sottodeterminato dei pesi di cubatura w_1, \dots, w_{η_n}

$$\sum_{i=1}^M w_i p_j(x_i) = \int_{\Omega} p_j(x) d\mu(x), \quad 1 \leq j \leq \eta_n,$$

che in forma matriciale si scrive

$$V^T \mathbf{w} = \mathbf{m}, \quad \mathbf{m} = (m_1, \dots, m_{\eta_n})^T, \quad m_j = \int_{\Omega} p_j(x) d\mu(x), \quad 1 \leq j \leq \eta_n \quad (4.6)$$

dove m_j , $j = 1, \dots, \eta_n$, sono i *momenti*. Supponendo i momenti noti, si può risolvere il sistema (4.6) tramite il metodo SVD e ricavare la soluzione \mathbf{w} , che in generale sarà costituita da M pesi w_i , $i = 1, \dots, M$, tutti non nulli.

Se si utilizza invece il comando “backslash” di Matlab, si ottengono esattamente η_n pesi non nulli, che determinano quindi anche gli η_n nodi rilevanti tra gli $M \geq \eta_n$ punti x_1, \dots, x_M di partenza. Ossia, i pesi di cubatura w_1, \dots, w_{η_n} possono essere calcolati assieme ai punti approssimati di Fekete $\{\xi_1, \dots, \xi_M\}$ all'interno dello stesso algoritmo AFP [5]. Prendendo infatti

$$\mathbf{b} = \mathbf{m} = \int_{\Omega} \mathbf{p}(x) d\mu(x)$$

nella formula (4.5), il vettore $\mathbf{w}(ind)$ conterrà i pesi per la formula di cubatura nei corrispondenti punti approssimati di Fekete $\mathbf{x}(ind)$.

Capitolo 5

Esperimenti numerici su domini 2D

In questo capitolo eseguiremo esperimenti numerici per alcuni domini e funzioni bivariate, mettendo a confronto i metodi di cubatura che hanno grado di esattezza polinomiale (qualora ce ne siano) con il metodo di integrazione quasi-Monte Carlo (d’ora in poi “QMC”), con la compressione del numero dei nodi quasi-random tramite `lsqnonneg`, e con quella sui punti approssimati di Fekete (d’ora in poi “AFP”).

La cubatura con il metodo QMC viene eseguita utilizzando la formula (1.10) generalizzata a domini Ω , ossia

$$I_{N_{qmc}}(f) := \frac{\lambda_d(\Omega)}{N_{qmc}} \sum_{i=1}^{N_{qmc}} f(x_i) \approx \int_{\Omega} f(x) dx, \quad (5.1)$$

dove $\{x_i\}_{1 \leq i \leq N_{qmc}}$ sono un insieme di punti quasi-random sul dominio Ω . L’insieme di punti quasi-random viene ricavato da una griglia quasi-random iniziale, di una certa cardinalità N , creata su un rettangolo con i lati allineati agli assi coordinati e tale che contenga il dominio. Dopodiché, tramite funzioni costruite ad hoc, vengono selezionati solo i punti che si trovano all’interno del dominio, che saranno in numero $N_{qmc} \leq N$, e a questi viene applicata la formula (5.1). Chiaramente, l’ideale sarebbe prendere il rettangolo circoscritto al dominio, ovvero una sorta di “bounding box”. Per i poligoni e il dominio composto, si è operato in questa maniera; invece in domini come la lente, per motivi di semplicità implementativa, ci si è accontentati di un rettangolo abbastanza aderente alla lente, anche se non perfettamente circoscritto. Le griglie iniziali vengono create un’unica volta all’inizio, mediante i comandi

```
P = haltonset(2);  
Pmc = net(P,N);
```

nel caso dei punti di Halton, e

```
P = sobolset(2);  
Pmc = net(P,N);
```

nel caso dei punti di Sobol, dove in entrambi i casi **Pmc** denota la griglia 2-dimensionale prodotta. In questo modo, i confronti tra i metodi QMC e di compressione mediante **lsqnonneg** e AFP vengono condotti sul medesimo insieme di punti.

Il metodo di compressione dei punti quasi-random tramite la function **lsqnonneg** è implementato dalla funzione Matlab **compresscub**, che si può trovare nel CD allegato al lavoro. Questa ha in input l'insieme di punti da comprimere, i relativi pesi, e l'opzione se richiedere che i nodi estratti abbiano pesi tutti positivi. È attraverso questa opzione che viene richiamata **lsqnonneg**. Tramite l'Algoritmo NNLS descritto nel Capitolo 3, **lsqnonneg** risolve il problema

$$\begin{aligned} \min_{\mathbf{w}} \quad & \|Q^T \mathbf{w} - \mathbf{m}\| \\ & w_i \geq 0 \quad i = 1, \dots, N_{qmc}, \end{aligned} \quad (5.2)$$

dove

- $Q^T \in \mathbb{R}^{\eta_n \times N_{qmc}}$ è la trasposta della matrice di Vandermonde V ortogonalizzata (per ridurre il malcondizionamento), calcolata sull'insieme di N_{qmc} punti iniziali quasi-random, nella base di Chebyshev $\{p_j\}_{1 \leq j \leq \eta_n}$, $\eta_n = \dim(\mathbb{P}_n^d) = \binom{n+d}{d}$, e tipicamente $\eta_n \ll N_{qmc}$;
- $\mathbf{m} = (m_1, \dots, m_{\eta_n})^T$ è il vettore dei momenti $m_i = \int_{\Omega} p_j(x) d(x)$, $1 \leq j \leq \eta_n$, nella base di Chebyshev $\{p_j\}_{1 \leq j \leq \eta_n}$;
- $\mathbf{w} \in \mathbb{R}_+^{N_{qmc}}$ è il vettore dei pesi da calcolare.

Tra i nuovi pesi calcolati, quelli strettamente positivi saranno chiaramente in numero minore od uguale a quello dei punti di partenza, $N_{pos} \leq N_{qmc}$, e determinano i nuovi nodi per la cubatura. I punti associati a pesi nulli, infatti, non danno contributo alla formula

$$I_{pos}(f) := \sum_{k=1}^{N_{pos}} w_k f(\xi_k) \approx \int_{\Omega} f(x) dx, \quad (5.3)$$

e vengono perciò scartati. Nel caso in cui *non* venga scelta l'opzione di avere pesi solo positivi o nulli, **compresscub** utilizza l'algoritmo QR con pivoting per colonne per ridurre il numero di nodi, che in Matlab è implementato dall'operatore "backslash". Quanto si è appena detto descrive il seguente frammento di codice di **compresscub**, dove **pts** sono i punti da comprimere; **Q** è la trasposta della matrice di Vandermonde V ortogonalizzata, calcolata nei punti **pts** e nella base di Chebyshev; **mm** sono i momenti nella base di Chebyshev ortogonalizzata; **pos** è l'opzione se richiedere che i nuovi pesi siano tutti positivi o nulli; **neww** sono i nuovi pesi prodotti da **lsqnonneg** o dalla fattorizzazione QR; **ind** sono gli indici dei nuovi pesi strettamente positivi; e **newpts** sono i nodi estratti dai punti **pts** associati ai pesi **neww** strettamente positivi.

```
% extracts the new nodes and weights
if pos == 1
    neww = lsqnonneg(Q',mm);
else
    neww = Q'\mm;
```

```

end;

% indexes of nonvanishing weights
ind = find(abs(neww)>0);

% compression
newpts = pts(ind,:);
neww = neww(ind);

```

I momenti che vengono dati in input a `lsqnonneg` sono calcolati da `compresscub` a partire dai punti e dai nodi iniziali che gli vengono passati, come si può vedere nel seguente frammento di codice

```

V = chebvand(deg,pts,rect);
% polynomial basis orthogonalization
[Q,R]=qr(V,0);
Q=real(Q);
% moments of the Chebyshev basis
mm = Q'*w;

```

dove V è la matrice di Vandermonde nella base di Chebyshev e calcolata nei punti `pts`; `deg` è il grado della base polinomiale di Chebyshev; `rect` è il più piccolo rettangolo con i lati allineati agli assi coordinati e contenente i punti `pts`; e w sono i pesi associati ai punti iniziali `pts`. Nel nostro caso, quindi, passandogli come nodi iniziali un insieme di punti quasi-random e come pesi iniziali i pesi QMC, $w_i = \lambda_d(\Omega)/N_{qmc}$, si ha che i momenti determinati da `compresscub` sono necessariamente approssimati (da cui li chiameremo “momenti quasi-Monte Carlo”). Tuttavia, qualora si disponga anche di nodi e pesi prodotti da un metodo con grado algebrico esatto, è possibile darli in input alla funzione `compresscub`, che provvederà a usare questi nodi e pesi per calcolare i momenti (che chiameremo “momenti esatti”¹), anziché quelli QMC. In questo modo, i nuovi nodi e pesi prodotti da `lsqnonneg` avranno precisione maggiore. Questo lo si ritrova nelle righe di codice

```

Vq = chebvand(deg,pts_quad,rect);
% polynomial basis orthogonalization
[Q,R] = qr(Vq,0);
Q = real(Q);
% moments of the orthogonal basis
mm = Q'*w_quad;

```

dove `pts_quad` e `w_quad` sono rispettivamente i nodi e i pesi prodotti da un metodo di quadratura con grado algebrico esatto; e V_q è la matrice di Vandermonde nella base di Chebyshev e calcolata nei punti `pts_quad`. Nella Tabella 5.1 sono riportati gli errori relativi (arrotondati alla quarta cifra decimale) dei momenti QMC rispetto ai momenti esatti per il poligono convesso, il poligono non convesso e la lente, che sono i domini per i quali disponiamo dei momenti esatti. Come si può osservare, gli errori sono abbastanza piccoli (dell'ordine di 10^{-4}), anche se tendono molto lentamente a crescere con il grado n .

¹Con leggero abuso di linguaggio, chiameremo “momenti esatti” i momenti calcolati a partire da nodi e pesi prodotti da un metodo con grado algebrico esatto. Qualora vi sia possibilità di fraintendimento con i *veri* momenti esatti si tornerà esplicitamente ad usare i termini propri.

Una stima dell'errore di cubatura per il metodo di compressione tramite `lsqnonneg` viene dal seguente risultato [58], che tiene conto anche dell'errore apportato dai momenti approssimati prodotti dal metodo `lsqnonneg`.

Teorema 14. *Dato un sottoinsieme compatto Ω di \mathbb{R}^d , si consideri la formula di cubatura $S_1(f)$ di grado n individuata da una misura ν multivariata, non necessariamente positiva, a supporto su un insieme finito $P = \{P_1, \dots, P_{N_1}\} \subset \Omega$, con rispettivi pesi $\mathbf{w}^T = \{w_1, \dots, w_{N_1}\}$*

$$S_1(f) = \int_{\Omega} f(P) d\nu = \sum_{i=1}^{N_1} w_i f(P_i). \quad (5.4)$$

Si consideri, inoltre, una seconda formula di cubatura discreta, con nodi $Q = \{Q_1, \dots, Q_{N_2}\} \subset \Omega$ e pesi $\boldsymbol{\omega}^T = \{\omega_1, \dots, \omega_{N_2}\}$

$$S_2(f) = \sum_{k=1}^{N_2} \omega_k f(Q_k). \quad (5.5)$$

Definiamo il vettore dei momenti discreti del metodo S_1

$$\mathbf{m} = V^T \mathbf{w} \quad (5.6)$$

e il vettore dei momenti discreti del metodo S_2

$$\boldsymbol{\mu} = W^T \boldsymbol{\omega}, \quad (5.7)$$

dove V è la matrice di Vandermonde $V = \{v_{ij}\} = \{\phi_j(P_i)\}$ nella base $\{\phi_j\}_{1 \leq j \leq \eta_n}$, $\eta_n = \dim(\mathbb{P}_n^d)$, calcolata nei punti di P ; e W è la matrice di Vandermonde $W = \{v_{kj}\} = \{\phi_j(Q_k)\}$ nella base $\{\phi_j\}_{1 \leq j \leq \eta_n}$, calcolata nei punti di Q . Definiamo inoltre l'errore assoluto dei momenti discreti

$$\varepsilon_{mom} = \|\mathbf{m} - \boldsymbol{\mu}\|_2. \quad (5.8)$$

Allora per ogni $f \in \mathcal{C}(\Omega)$ vale la seguente stima dell'errore di cubatura di $S_2(f)$ rispetto a $S_1(f)$

$$\begin{aligned} & \left| \int_{\Omega} f(P) d\nu - \sum_{k=1}^{N_2} \omega_k f(Q_k) \right| \\ & \leq \left(\nu(\Omega) + \sum_{k=1}^{N_2} |\omega_k| + \sqrt{\eta(\Omega)} \varepsilon_{mom} \right) E_n(f; \Omega) + \|f\|_{L_{d\lambda}^2(\Omega)} \varepsilon_{mom} \\ & = \mathcal{O}(E_n(f; \Omega)) + \mathcal{O}(\varepsilon_{mom}) \end{aligned} \quad (5.9)$$

dove η è la misura a supporto su Ω rispetto alla quale è definito il prodotto scalare per la base $\{\phi_j\}_{1 \leq j \leq \eta_n}$, e $E_n(f; \Omega)$ è l'errore di miglior approssimazione polinomiale uniforme di grado al più n per f su Ω , ossia

$$E_n(f; \Omega) = \min_{p \in \mathbb{P}_n^d} \|f - p\|_{L^\infty(\Omega)} = \|f - p^*\|_{L^\infty(\Omega)}$$

con p^ polinomio di miglior approssimazione uniforme di grado al più n per f su Ω .*

	$n = 10$	$n = 20$	$n = 30$
Poligono convesso	1.1870e-04	2.3670e-04	3.7318e-04
Poligono non convesso	1.7300e-04	3.1276e-04	4.8967e-04
Lente	1.3258e-04	2.6496e-04	4.1986e-04

Tabella 5.1: Errori relativi dei momenti QMC rispetto ai momenti esatti.

Ora, tenendo presente il Teorema 14, supponiamo che

- ν sia la misura discreta che ha come pesi w_i i pesi di una certa formula $S_1(f)$ con grado algebrico esatto (ad esempio quella di `Polygauss` o `gqlens`);
- P sia l'insieme dei nodi determinato da tale metodo di cubatura con grado algebrico esatto;
- Q sia l'insieme dei punti estratti da `lsqnonneg` da un insieme di punti quasi-random;
- ω_k siano i pesi calcolati da `lsqnonneg`,

allora la relazione (5.9) diventa la stima dell'errore assoluto di cubatura commesso dalla compressione tramite `lsqnonneg` rispetto al metodo con grado algebrico esatto $S_1(f)$. Se invece operiamo la seguente scelta di ν e P

- ν sia la misura discreta di Monte Carlo su Ω , con pesi $w_i = \lambda(\Omega)/N_{qmc}$, $1 \leq i \leq N_{qmc}$;
- $P = \{P_1, \dots, P_{N_{qmc}}\}$ sia un insieme di punti quasi-random,

allora si ottiene la stima dell'errore assoluto di cubatura commesso da `lsqnonneg` rispetto al metodo di integrazione QMC. Analogo discorso se si sceglie di mettere a confronto il metodo di integrazione QMC con un metodo con grado algebrico esatto. In entrambi i casi, la stima (5.9) assicura di fatto la convergenza del metodo di compressione `lsqnonneg` rispettivamente al metodo con grado algebrico esatto e al metodo di integrazione QMC, a meno di un errore dipendente dai momenti.

Gli AFP vengono estratti dall'insieme iniziale di punti quasi-random tramite la routine di Matlab `approxfek` [56, 4], che implementa l'Algoritmo AFP Greedy descritto nel Capitolo 4. Anche in questo caso, se si dispone dei momenti esatti sul dominio, si calcolano i pesi esatti per i nodi AFP, altrimenti si possono calcolare i momenti QMC in maniera analoga a quanto fatto in `compresscube`, ottenendo pesi approssimati per i nodi AFP. In entrambi i casi, i pesi calcolati possono essere in generale negativi. Riportiamo le righe di codice che calcolano gli AFP (`fek`) da un insieme di punti quasi-random (`Pmc_in`), i rispettivi pesi (`w_fek`) e l'integrale sugli AFP (`I_fek`).

```
% Estrazione degli AFP
[fek,R1,R2,leb] = approxfek(deg,Pmc_in,leja,controlmesh,rect);

% Calcolo dei pesi
w_fek = chebvand(deg, fek, rect);
```

```

w_fek = V_fek'\m; % pesi degli AFP

% Calcolo dell'integrale
X = fek(:,1); Y = fek(:,2);
f_fek = feval(f,X,Y); % funzione test valutata negli AFP
I_fek = w\_fek'*f_fek; % integrale calcolato sugli AFP

```

Naturalmente, la stima (5.9) può essere fatta anche per la cubatura sugli AFP rispetto ai metodi sopra visti.

Nelle prossime sezioni presenteremo i risultati degli esperimenti numerici ottenuti applicando questi metodi a quattro domini, ossia un poligono convesso, un poligono non convesso, una lente e un dominio composito, e ad altrettante funzioni test:

$$f_1(x, y) = \frac{3}{4}e^{-\frac{1}{4}((9x-2)^2+(9y-2)^2)} + \frac{3}{4}e^{-\frac{1}{49}(9x+1)^2-\frac{1}{10}(9y+1)^2} + \frac{1}{2}e^{-\frac{1}{4}((9x-7)^2+(9y-3)^2)} - \frac{1}{5}e^{-(9x-4)^2-(9y-7)^2} \quad (5.10)$$

$$f_2(x, y) = \sqrt{(x-0.5)^2 + (y-0.5)^2} \quad (5.11)$$

$$f_3(x, y) = e^{-100((x-0.5)^2+(y-0.5)^2)} \quad (5.12)$$

$$f_4(x, y) = \cos(30(x+y)). \quad (5.13)$$

Anche nel caso di domini per i quali si disponga di momenti esatti, effettueremo la cubatura con `lsqnonneg` e con gli AFP sia con i momenti esatti sia con momenti QMC. Il motivo è che confrontare i risultati ottenuti da momenti esatti e da momenti approssimati permette di stabilire quanto possa essere ragionevole - in termini di buona approssimazione - usare i momenti QMC in mancanza di momenti esatti.

Per ogni dominio e funzione test, verranno condotti esperimenti su tre differenti gradi polinomiali, $n = 10, 20, 30$, e su tre griglie di punti quasi-random, rispettivamente di cardinalità $N = 50000, 100000, 200000$. I punti quasi-random utilizzati sono i punti di Halton e di Sobol. Occorre tenere presente che i vari metodi presi in considerazione dipendono in maniera diversa dal grado n e dalla cardinalità della griglia di partenza N , nello specifico:

- i metodi con grado di esattezza polinomiale `Polygauss` e `gqlens` dipendono solamente da n ;
- l'integrazione QMC dipende soltanto da N ;
- il metodo di compressione con `lsqnonneg` dipende sia da n sia da N ; in particolare, sia il valore degli integrali sia il numero dei nodi estratti da `lsqnonneg` dipendono da n e N ;
- il calcolo dei nodi AFP dipende sia da n sia da N . Tuttavia, mentre gli integrali calcolati sugli AFP sono determinati sia da n sia da N , il numero dei nodi dipende soltanto da n , essendo uguale alla dimensione dello spazio polinomiale \mathbb{P}_n^2 , ovvero $(n+1)(n+2)/2$.

Per il poligono convesso, non convesso e la lente, verranno presentati gli errori relativi compiuti dal metodo d'integrazione QMC, dalla compressione via

`lsqnonneg` con momenti esatti e con momenti QMC, e dalla cubatura condotta sugli AFP con momenti esatti e con momenti QMC. I valori di cubatura presi come base di confronto sono quelli ottenuti, a pari grado n , dai metodi `Polygauss` (per i poligoni) e `gqlens` (per la lente).

Nel caso del dominio composito per il quale, in virtù della difficile geometria, non sono noti metodi esatti, si assumono come valori di riferimento i risultati del metodo QMC; gli errori relativi calcolati sono quelli prodotti dalla compressione via `lsqnonneg` con momenti QMC e dalla cubatura sugli AFP con momenti quasi-Monte Carlo.

È importante osservare che gli integrali che vengono presi come valori di riferimento sono comunque approssimati. Inoltre il confronto di ciascun metodo rispetto ai risultati del metodo di riferimento viene condotto *con i medesimi n e N* . Nel caso dei poligoni, per ogni grado n viene considerato come integrale di riferimento quello calcolato da `Polygauss` proprio al grado n (e quindi non si è usato `Polygauss` con un grado più elevato $\hat{n} \gg n$ per avere un risultato di cubatura più vicino a quello esatto). Analogamente con `gqlens` per la lente. Per il dominio composito vale un discorso simile: si è scelta l'integrazione QMC come metodo di riferimento, e la si è usata per produrre integrali con lo stesso valore N degli altri metodi, non con un valore superiore $\hat{N} \gg N$. Ovvero, l'analisi degli errori è stata condotta rispetto al (comportamento del) metodo preso come riferimento, non rispetto all'integrale esatto vero e proprio.

Verrà inoltre riportata la tabella con il numero di nodi impiegati da ciascun metodo, al variare del grado n e dal numero di punti N della griglia quasi-random iniziale. Inoltre, nel caso di pesi potenzialmente negativi, viene calcolato il rapporto

$$\rho = \frac{\sum_i |w_i|}{|\sum_i w_i|} \quad (5.14)$$

che assume valori nell'intervallo $[1, +\infty[$. La quantità ρ registra l'entità dei pesi di segno negativo rispetto a quelli positivi, in quanto $\rho = 1$ se i pesi sono tutti positivi; tale valore può essere considerato un parametro di stabilità, poiché, come noto, la presenza di pesi di segno negativo può favorire fenomeni di instabilità.

Tutti gli esperimenti numerici sono stati realizzati con Matlab 7.10.0, su un 3.00 GHz Intel Core 2 Duo con 4 GB di RAM. I tempi di esecuzione di `Polygauss` e del metodo QMC sono inferiori al secondo, mentre quelli di `lsqnonneg` e di `approxfek` vanno da pochi secondi ($n = 10$, $N = 50000$) a un paio di minuti ($n = 30$, $N = 200000$).

5.1 Poligono convesso

Il poligono convesso che prendiamo in considerazione ha vertici $P_1 = (1, 0)$, $P_2 = (4, 0)$, $P_3 = (7, 2)$, $P_4 = (8, 3)$, $P_5 = (7, 5)$, $P_6 = (4, 7)$, $P_7 = (1, 6)$, $P_8 = (0, 3)$, $P_9 = (0, 2)$. Le griglie quasi-random iniziali sono costituite da punti di Halton o di Sobol. I momenti esatti sono stati calcolati usando nodi e pesi prodotti da **Polygauss**, mentre i momenti QMC sono stati determinati a partire da una griglia iniziale di 600000 punti di Halton.

Nella Figura 5.1 si può osservare la distribuzione dei nodi prodotti da **Polygauss** e da **lsqnonneg**, e degli AFP. Questi ultimi due insiemi di punti, in particolare, sono disposti in maniera omogenea su tutto il dominio, accumulandosi verso il bordo.

Nelle Tabelle 5.2 e 5.3 è riportato il numero di nodi usati da ciascun metodo, al variare del grado n e della cardinalità N della griglia quasi-random iniziale (rispettivamente di Halton e di Sobol). Il numero dei nodi prodotti da **Polygauss** è notevolmente maggiore di quello degli AFP. Il numero dei nodi determinati da **lsqnonneg**, invece, è quasi sempre uguale al numero degli AFP, tranne che per i nodi estratti con momenti esatti, il cui numero risulta minore per i gradi $n = 20$ e, soprattutto, $n = 30$. I rapporti (5.14) riportati in tabella sono arrotondati alla seconda cifra decimale, e si dimostrano ottimali per i pesi prodotti da **Polygauss**, e abbastanza buoni per i pesi degli AFP eccetto che per quelli calcolati con momenti QMC al grado $n = 30$.

Nella Tabella 5.4 sono elencati i risultati della cubatura ottenuta dal metodo

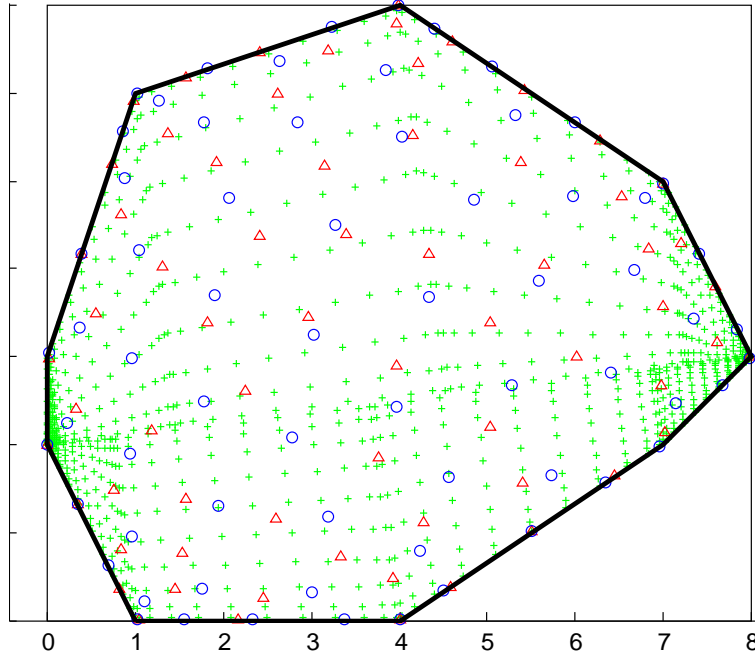


Figura 5.1: Poligono convesso con $N = 200000$, $n = 10$: distribuzione dei punti prodotti dal metodo **Polygauss** (+), dei punti selezionati dalla griglia di Halton tramite **lsqnonneg** con momenti esatti (Δ), e degli AFP (\circ).

con grado algebrico esatto **Polygauss**. Gli integrali calcolati con le funzioni integrande f_3 ed f_4 mostrano differenze importanti al variare del grado n , in particolare f_4 . Ci si aspetta, quindi, che per le funzioni f_3 ed f_4 gli errori relativi dei vari metodi rispetto a **Polygauss** siano grandi, essendo i risultati stessi di **Polygauss** non esatti.

Riportiamo nelle Tabelle 5.5-5.8 e 5.9-5.12 gli errori relativi commessi dai vari metodi rispetto ai risultati di **Polygauss** (Tabella 5.4), rispettivamente per i punti di Halton e i punti di Sobol. Gli errori sono particolarmente elevati per le funzioni f_3 e, soprattutto, f_4 . Come già accennato, ci si aspettava un risultato simile, poiché i valori degli integrali di f_3 e f_4 calcolati con **Polygauss** e rispetto ai quali vengono calcolati gli errori relativi sono già in partenza molto variabili. Gli errori per le funzioni f_1 e, soprattutto, f_2 sono invece in generale piccoli. Questo trova una possibile spiegazione nel Teorema di Koksma-Hlawka (Teorema 5), che afferma che la bontà dei risultati QMC può dipendere in generale dalla variazione della funzione integranda sul dominio. Difatti, la funzione che si comporta meglio con tutti questi metodi - che sono basati su punti QMC - è f_2 , che ha variazione piccola sul dominio.

Si possono fare alcune osservazioni riguardo agli errori commessi dai vari metodi, che però non valgono per le funzioni test f_3 e f_4 , i cui errori sono di grande entità e hanno un andamento imprevedibile per il motivo visto sopra.

Gli errori ottenuti dall'integrazione QMC chiaramente diminuiscono all'aumentare della cardinalità N della griglia quasi-random di partenza.

Gli errori dati dal metodo di compressione **lsqnonneg** con momenti esatti mostrano un significativo miglioramento all'aumentare del grado n , mentre quelli prodotti con momenti QMC tendono a migliorare più che altro all'aumentare della cardinalità N della griglia quasi-random iniziale.

Quanto alla cubatura sugli AFP, nel caso di f_1 gli integrali migliorano sia all'aumentare di n sia all'aumentare della cardinalità della griglia quasi-random iniziale. I risultati ricavati da momenti esatti, in particolare, registrano un miglioramento significativo più al crescere del grado n che della cardinalità N . Comportamento analogo per la funzione f_2 , dove però l'errore di cubatura sugli AFP con momenti QMC rimane pressoché uguale (ma comunque molto buono) al variare di n e N .

In generale, si può vedere che per gradi bassi (almeno fino al grado $n = 20$ per f_1 , e fino al grado $n = 10$ per f_2) il metodo QMC si dimostra equivalente o addirittura migliore rispetto agli altri metodi. Per gradi più alti, invece, viene sorpassato da **lsqnonneg** e dall'integrazione sugli AFP con momenti esatti. Utilizzare **lsqnonneg** e gli AFP con momenti QMC pur disponendo dei momenti esatti non è sensato, visto che forniscono entrambi risultati peggiori rispetto a quelli con i momenti esatti; tuttavia, è interessante notare che i risultati ottenuti con momenti QMC sono comunque abbastanza buoni, in particolare gli integrali sugli AFP, che per gradi sufficientemente alti ($n = 30$ per f_1 , $n = 20$ per f_2) arrivano ad essere migliori del metodo QMC.

Infine, dalle tabelle del numero dei nodi 5.2-5.3 e degli errori 5.5-5.12, si può vedere che i punti di Sobol si comportano in maniera molto simile a quella dei Halton. Per motivi di brevità, quindi, nei prossimi domini continueremo ad usare solo i punti di Halton.

n	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
10	Polygauss	990 (1.00)	990 (1.00)	990 (1.00)
	QMC	35266	70524	141065
	lsqnonneg m. esatti	66	66	66
	lsqnonneg m. QMC	66	66	66
	AFP m. esatti	66 (1.01)	66 (1.03)	66 (1.03)
	AFP m. QMC	66 (1.01)	66 (1.03)	66 (1.03)
20	Polygauss	3780 (1.00)	3780 (1.00)	3780 (1.00)
	QMC	35266	70524	141065
	lsqnonneg m. esatti	209	210	205
	lsqnonneg m. QMC	231	231	231
	AFP m. esatti	231 (1.01)	231 (1.03)	231 (1.01)
	AFP m. QMC	231 (1.01)	231 (1.03)	231 (1.01)
30	Polygauss	8370 (1.00)	8370 (1.00)	8370 (1.00)
	QMC	35266	70524	141065
	lsqnonneg m. esatti	407	405	397
	lsqnonneg m. QMC	496	496	494
	AFP m. esatti	496 (1.02)	496 (1.06)	496 (1.01)
	AFP m. QMC	496 (1.10)	496 (1.14)	496 (1.20)

Tabella 5.2: Numero dei nodi sul poligono convesso prodotti da **Polygauss**, dei punti quasi-random contenuti nel dominio, dei punti estratti da **lsqnonneg** con momenti esatti/QMC, e degli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / |\sum w_i|$.

n	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
10	Polygauss	990 (1.00)	990 (1.00)	990 (1.00)
	QMC	35273	70544	141090
	lsqnonneg m. esatti	66	66	65
	lsqnonneg m. QMC	66	66	66
	AFP m. esatti	66 (1.01)	66 (1.03)	66 (1.03)
	AFP m. QMC	66 (1.01)	66 (1.03)	66 (1.03)
20	Polygauss	3780 (1.00)	3780 (1.00)	3780 (1.00)
	QMC	35273	70544	141090
	lsqnonneg m. esatti	209	206	204
	lsqnonneg m. QMC	231	231	231
	AFP m. esatti	231 (1.01)	231 (1.03)	231 (1.02)
	AFP m. QMC	231 (1.01)	231 (1.03)	231 (1.02)
30	Polygauss	8370 (1.00)	8370 (1.00)	8370 (1.00)
	QMC	35273	70544	141090
	lsqnonneg m. esatti	407	403	400
	lsqnonneg m. QMC	496	496	495
	AFP m. esatti	496 (1.01)	496 (1.04)	496 (1.03)
	AFP m. QMC	496 (1.22)	496 (1.18)	496 (1.19)

Tabella 5.3: Numero dei nodi sul poligono convesso prodotti da **Polygauss**, dei punti quasi-random contenuti nel dominio, dei punti estratti da **lsqnonneg** con momenti esatti/QMC, e degli AFP momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Sobol iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / \sum w_i$.

$n = 10$	1.8371459710069199e-01	$n = 10$	1.7910732800264310e+02
f_1 $n = 20$	1.8376559061448833e-01	f_2 $n = 20$	1.7910732613130250e+02
$n = 30$	1.8376554711203447e-01	$n = 30$	1.7910732613752995e+02
$n = 10$	1.9686138093479219e-05	$n = 10$	1.9219850008717834e+00
f_3 $n = 20$	2.4825391507378604e-05	f_4 $n = 20$	-7.4881395854254595e-01
$n = 30$	2.4590157563811153e-05	$n = 30$	1.9901702982163538e-03

Tabella 5.4: Valori degli integrali ottenuti dal metodo **Polygauss** sul poligono convesso, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	1.8e-03	1.6e-03	3.8e-04
	lsqnonneg m. esatti	6.0e-02	1.5e-01	3.0e-02
	lsqnonneg m. QMC	3.4e-01	1.9e-02	1.5e-02
	AFP m. esatti	1.6e-01	4.1e-02	8.3e-02
	AFP m. QMC	1.6e-01	4.2e-02	8.4e-02
$n = 20$	QMC	1.5e-03	1.8e-03	6.6e-04
	lsqnonneg m. esatti	2.2e-02	8.2e-03	2.6e-02
	lsqnonneg m. QMC	3.0e-02	2.7e-02	3.5e-03
	AFP m. esatti	2.7e-02	1.6e-02	2.8e-02
	AFP m. QMC	2.6e-02	1.5e-02	2.8e-02
$n = 30$	QMC	1.5e-03	1.8e-03	6.6e-04
	lsqnonneg m. esatti	4.3e-04	3.3e-03	9.3e-03
	lsqnonneg m. QMC	1.7e-03	7.5e-03	4.5e-03
	AFP m. esatti	6.1e-04	2.9e-03	4.5e-03
	AFP m. QMC	2.3e-04	1.0e-02	3.0e-02

Tabella 5.5: Errori relativi degli integrali di f_1 sul poligono convesso, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `Polygauss`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	1.7e-04	2.8e-04	3.3e-05
	lsqnonneg m. esatti	1.8e-05	1.1e-04	7.7e-05
	lsqnonneg m. QMC	2.7e-04	2.5e-04	5.1e-05
	AFP m. esatti	3.9e-05	1.3e-04	1.1e-04
	AFP m. QMC	6.6e-05	1.1e-04	1.3e-04
$n = 20$	QMC	1.7e-04	2.8e-04	3.3e-05
	lsqnonneg m. esatti	1.1e-08	3.0e-06	1.2e-06
	lsqnonneg m. QMC	1.7e-04	2.7e-04	3.4e-05
	AFP m. esatti	4.1e-06	1.1e-05	4.1e-06
	AFP m. QMC	2.3e-05	1.7e-05	2.3e-05
$n = 30$	QMC	1.7e-04	2.8e-04	3.3e-05
	lsqnonneg m. esatti	3.7e-07	1.7e-06	2.1e-07
	lsqnonneg m. QMC	1.7e-04	2.8e-04	3.2e-05
	AFP m. esatti	2.4e-07	5.6e-07	1.2e-06
	AFP m. QMC	2.7e-05	3.0e-05	2.1e-05

Tabella 5.6: Errori relativi degli integrali di f_2 sul poligono convesso, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `Polygauss`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	1.8e-01	3.1e-01	2.6e-01
	lsqnonneg m. esatti	9.1e-01	1.0e+00	9.8e-01
	lsqnonneg m. QMC	1.0e+00	1.0e+00	1.0e+00
	AFP m. esatti	5.8e-01	3.9e+00	5.5e+01
	AFP m. QMC	5.8e-01	3.8e+00	5.5e+01
$n = 20$	QMC	6.5e-02	4.3e-02	4.3e-03
	lsqnonneg m. esatti	2.6e+00	3.5e+00	9.8e-01
	lsqnonneg m. QMC	9.3e-02	1.0e+00	9.5e-01
	AFP m. esatti	8.0e-01	9.6e-01	6.2e-01
	AFP m. QMC	8.0e-01	9.6e-01	6.2e-01
$n = 30$	QMC	5.6e-02	5.3e-02	5.2e-03
	lsqnonneg m. esatti	8.7e-01	6.7e-01	6.9e-01
	lsqnonneg m. QMC	3.3e-01	7.5e-01	2.6e+00
	AFP m. esatti	2.5e-02	4.8e+00	4.6e-01
	AFP m. QMC	3.2e-01	1.2e+01	4.7e+00

Tabella 5.7: Errori relativi degli integrali di f_3 sul poligono convesso, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `Polygauss`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	9.7e-01	9.7e-01	9.9e-01
	lsqnonneg m. esatti	7.6e-01	2.4e+00	1.8e+00
	lsqnonneg m. QMC	3.4e+00	1.4e+00	2.4e-01
	AFP m. esatti	2.6e+00	3.0e+00	1.9e+00
	AFP m. QMC	2.6e+00	3.0e+00	1.9e+00
$n = 20$	QMC	1.1e+00	1.1e+00	1.0e+00
	lsqnonneg m. esatti	5.9e+00	2.3e+00	9.8e-01
	lsqnonneg m. QMC	4.7e+00	4.9e-01	1.9e+00
	AFP m. esatti	3.9e-01	1.6e-01	4.5e+00
	AFP m. QMC	3.9e-01	1.6e-01	4.5e+00
$n = 30$	QMC	2.6e+01	2.4e+01	7.1e+00
	lsqnonneg m. esatti	2.5e+01	1.0e+03	7.3e+02
	lsqnonneg m. QMC	1.2e+02	1.1e+03	2.0e+02
	AFP m. esatti	1.4e+03	4.3e+02	1.4e+03
	AFP m. QMC	2.0e+03	3.6e+02	6.0e+02

Tabella 5.8: Errori relativi degli integrali di f_4 sul poligono convesso, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `Polygauss`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	2.3e-03	1.0e-03	3.5e-04
	lsqnonneg m. esatti	1.6e-01	1.0e-01	8.7e-02
	lsqnonneg m. QMC	1.8e-01	2.8e-02	1.1e-01
	AFP m. esatti	1.6e-01	1.6e-01	2.4e-02
	AFP m. QMC	1.6e-01	1.6e-01	2.5e-02
$n = 20$	QMC	2.6e-03	7.6e-04	7.0e-05
	lsqnonneg m. esatti	5.8e-03	6.4e-02	4.0e-02
	lsqnonneg m. QMC	2.9e-02	2.0e-02	2.3e-02
	AFP m. esatti	1.6e-02	1.6e-02	4.2e-03
	AFP m. QMC	1.7e-02	1.6e-02	4.7e-03
$n = 30$	QMC	2.6e-03	7.6e-04	7.0e-05
	lsqnonneg m. esatti	3.6e-04	7.0e-03	4.5e-03
	lsqnonneg m. QMC	2.6e-03	3.2e-03	6.9e-03
	AFP m. esatti	9.2e-03	1.8e-02	2.0e-03
	AFP m. QMC	1.7e-02	7.7e-02	7.9e-03

Tabella 5.9: Errori relativi degli integrali di f_1 sul poligono convesso, ottenuti dal metodo QMC su punti di Sobol, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Sobol iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo **Polygauss**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	1.7e-04	1.5e-04	1.8e-04
	lsqnonneg m. esatti	5.4e-05	1.4e-05	8.9e-06
	lsqnonneg m. QMC	3.0e-05	1.2e-04	3.9e-04
	AFP m. esatti	9.9e-05	2.9e-05	4.9e-05
	AFP m. QMC	7.2e-05	5.7e-05	7.6e-05
$n = 20$	QMC	1.7e-04	1.5e-04	1.8e-04
	lsqnonneg m. esatti	5.6e-06	2.7e-06	1.1e-06
	lsqnonneg m. QMC	1.7e-04	1.5e-04	1.7e-04
	AFP m. esatti	3.1e-07	6.3e-07	1.6e-07
	AFP m. QMC	2.7e-05	2.8e-05	2.7e-05
$n = 30$	QMC	1.7e-04	1.5e-04	1.8e-04
	lsqnonneg m. esatti	4.9e-07	5.9e-07	1.2e-07
	lsqnonneg m. QMC	1.7e-04	1.5e-04	1.8e-04
	AFP m. esatti	8.7e-07	3.9e-06	2.3e-07
	AFP m. QMC	2.7e-05	1.3e-05	2.6e-05

Tabella 5.10: Errori relativi degli integrali di f_2 sul poligono convesso, ottenuti dal metodo QMC su punti di Sobol, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Sobol iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo **Polygauss**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	4.5e-01	3.2e-01	3.0e-01
	lsqnonneg m. esatti	8.8e-01	1.0e+00	1.0e+00
	lsqnonneg m. QMC	1.0e+00	3.2e+01	1.0e+00
	AFP m. esatti	5.0e-01	5.4e-01	8.8e-01
	AFP m. QMC	5.0e-01	5.4e-01	8.8e-01
$n = 20$	QMC	1.5e-01	4.9e-02	2.7e-02
	lsqnonneg m. esatti	2.2e+00	3.7e+00	9.7e-01
	lsqnonneg m. QMC	8.5e-01	4.2e+00	9.8e-01
	AFP m. esatti	5.4e-01	4.7e-01	1.6e+00
	AFP m. QMC	5.4e-01	4.7e-01	1.6e+00
$n = 30$	QMC	1.6e-01	5.9e-02	3.7e-02
	lsqnonneg m. esatti	6.3e-01	2.4e-01	5.9e-01
	lsqnonneg m. QMC	6.8e-01	3.5e-01	1.1e+00
	AFP m. esatti	2.9e-01	9.3e+00	3.4e+00
	AFP m. QMC	4.4e-01	3.0e+01	4.2e+00

Tabella 5.11: Errori relativi degli integrali di f_3 sul poligono convesso, ottenuti dal metodo QMC su punti di Sobol, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Sobol iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo **Polygauss**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	9.9e-01	9.9e-01	1.0e+00
	lsqnonneg m. esatti	2.2e+00	1.1e-01	3.0e-01
	lsqnonneg m. QMC	1.9e+00	2.2e+00	3.2e-01
	AFP m. esatti	5.0e+00	1.7e+00	6.1e-01
	AFP m. QMC	5.0e+00	1.7e+00	6.1e-01
$n = 20$	QMC	1.0e+00	1.0e+00	1.0e+00
	lsqnonneg m. esatti	1.8e-01	2.7e+00	6.6e+00
	lsqnonneg m. QMC	6.3e+00	8.4e-01	1.5e+00
	AFP m. esatti	6.6e-01	5.9e+00	1.9e-01
	AFP m. QMC	6.6e-01	5.9e+00	1.9e-01
$n = 30$	QMC	6.1e+00	6.4e+00	1.4e+00
	lsqnonneg m. esatti	3.8e+02	3.0e+01	1.2e+03
	lsqnonneg m. QMC	3.2e+02	1.5e+03	1.4e+03
	AFP m. esatti	3.3e+02	8.3e+02	2.7e+02
	AFP m. QMC	1.3e+03	6.9e+02	1.9e+02

Tabella 5.12: Errori relativi degli integrali di f_4 sul poligono convesso, ottenuti dal metodo QMC su punti di Sobol, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Sobol iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo **Polygauss**.

5.2 Poligono non convesso

Il poligono non convesso che abbiamo scelto ha vertici $P_1 = (0, 1)$, $P_2 = (1, 0)$, $P_3 = (2, 0.5)$, $P_4 = (3, 0)$, $P_5 = (6.5, 0.5)$, $P_6 = (7, 2)$, $P_7 = (5, 4)$, $P_8 = (4, 3)$, $P_9 = (2.5, 3.5)$, $P_{10} = (1.5, 3.5)$. Le configurazioni iniziali utilizzate sono composte da punti di Halton. Anche in questo caso, i momenti esatti sono stati calcolati usando nodi e pesi prodotti da **Polygauss**, mentre i momenti QMC sono stati determinati a partire da una griglia iniziale di 600000 punti di Halton.

Nella Figura 5.2 sono raffigurati i nodi prodotti da **Polygauss** e da **lsqnonneg**, e gli AFP. Sia i punti estratti da **lsqnonneg** sia gli AFP hanno distribuzione omogenea su tutto il dominio, accumulandosi verso il bordo.

Nella Tabella 5.13 è riportato il numero di nodi usati da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Anche per il poligono non convesso il numero dei nodi usati da **Polygauss** è parecchio maggiore di quello degli AFP. Il numero dei nodi determinati da **lsqnonneg**, invece, si mantiene quasi sempre uguale al numero degli AFP con l'eccezione dei nodi estratti con momenti esatti, che risultano di meno per i gradi $n = 20$ e, soprattutto, $n = 30$. I rapporti dei pesi (5.14) sono ottimali per **Polygauss**, e comunque molto buoni anche per i pesi degli AFP, tranne che per quelli calcolati al grado $n = 10$ dalla griglia iniziale di $N = 200000$ punti di Halton.

I risultati della cubatura ottenuta da **Polygauss** sono elencati nella Tabella 5.14. Anche per questo poligono, gli integrali calcolati con la funzione integranda f_4 mostrano differenze importanti al variare del grado n , mentre l'integrale su f_3 ha una leggera variazione solo al grado $n = 10$. Nuovamente, si prevedono errori grandi per la funzione f_4 da parte dei vari metodi rispetto a **Polygauss**,

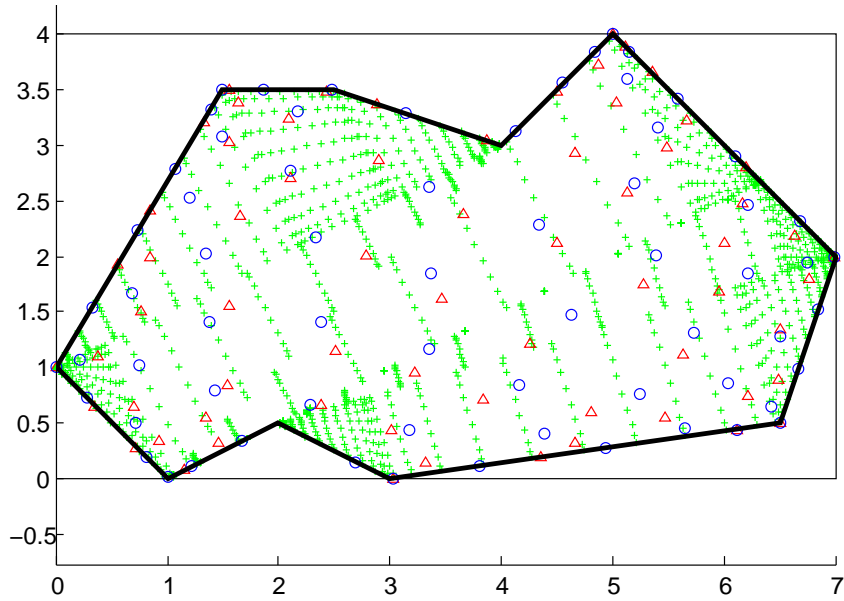


Figura 5.2: Poligono non convesso con $N = 200000$, $n = 10$: distribuzione dei punti prodotti dal metodo **Polygauss** (+), dei punti selezionati dalla griglia di Halton tramite **lsqnonneg** con momenti esatti (Δ), e degli AFP (o).

perché i risultati di **Polygauss** sono già in partenza non precisi.

Nelle Tabelle 5.15-5.18 sono riportati gli errori relativi degli integrali di ciascun metodo di cubatura rispetto ai risultati di **Polygauss** (Tabella 5.14). Come previsto, gli errori sono particolarmente rilevanti per la funzione f_4 e poco soddisfacenti per f_3 . Gli errori per le funzioni f_1 e, soprattutto, f_2 sono anche qui piuttosto piccoli; questo fatto, come già spiegato nel caso del poligono convesso, è in parte imputabile al Teorema di Koksma-Hlawka (Teorema 5).

Questa volta, gli errori ottenuti dall'integrazione QMC rimangono pressoché invariati all'aumentare della cardinalità N della griglia di Halton di partenza, con qualche piccola eccezione per f_3 . Gli errori dati dal metodo di compressione **lsqnonneg** con momenti esatti, invece, tendono a migliorare all'aumentare del grado n , mentre quelli prodotti con momenti QMC non registrano particolari cambiamenti, né al variare di n né al variare di N ; questo perché la cubatura tramite **lsqnonneg** con momenti QMC è fortemente influenzata dal comportamento dell'integrazione QMC. Quanto alla cubatura sugli AFP, nel caso di f_1 gli integrali migliorano all'aumentare del grado n della griglia iniziale, mentre per tutte le altre funzioni cambiano poco o nulla.

Questa volta l'integrazione QMC sulla funzione f_1 risulta sempre migliore rispetto agli altri metodi, e analogamente per la funzione test f_3 , i cui errori relativi, tuttavia, sono inficiati dai risultati poco precisi di **Polygauss** con cui viene fatto il confronto. Nel caso della funzione f_2 , invece, il metodo QMC dà in generale risultati uguali ($n = 10$) o peggiori ($n \geq 20$). Le cubature tramite **lsqnonneg** e sugli AFP realizzate con momenti esatti si dimostrano pressoché equivalenti.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	Polygauss	1430 (1.00)	1430 (1.00)	1430 (1.00)
	QMC	33488	66988	133936
	lsqnonneg m. esatti	66	66	66
	lsqnonneg m. QMC	66	66	66
	AFP m. esatti	66 (1.04)	66 (1.05)	66 (1.14)
	AFP m. QMC	66 (1.04)	66 (1.05)	66 (1.14)
$n = 20$	Polygauss	5460 (1.00)	5460 (1.00)	5460 (1.00)
	QMC	33488	66988	133936
	lsqnonneg m. esatti	210	205	206
	lsqnonneg m. QMC	231	231	231
	AFP m. esatti	231 (1.02)	231 (1.02)	231 (1.01)
	AFP m. QMC	231 (1.02)	231 (1.02)	231 (1.01)
$n = 30$	Polygauss	12090 (1.00)	12090 (1.00)	12090 (1.00)
	QMC	33488	66988	133936
	lsqnonneg m. esatti	406	402	396
	lsqnonneg m. QMC	496	494	495
	AFP m. esatti	496 (1.03)	496 (1.04)	496 (1.02)
	AFP m. QMC	496 (1.03)	496 (1.05)	496 (1.03)

Tabella 5.13: Numero dei nodi sul poligono non convesso prodotti da **Polygauss**, dei punti quasi-random contenuti nel dominio), dei punti estratti da **lsqnonneg** con momenti esatti/QMC, e degli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / \sum w_i$.

$n = 10$	1.9880236964884296e-01	$n = 10$	6.5499620016103009e+01
f_1 $n = 20$	1.9888114025520601e-01	f_2 $n = 20$	6.5499618791179472e+01
$n = 30$	1.9888118672674096e-01	$n = 30$	6.5499619605070308e+01
$n = 10$	1.5673198413802270e-02	$n = 10$	6.4835794807290403e-02
f_3 $n = 20$	1.5707980658347179e-02	f_4 $n = 20$	-1.3397145708172598e-01
$n = 30$	1.5707963298365771e-02	$n = 30$	2.1655590721568960e-02

Tabella 5.14: Valori degli integrali ottenuti dal metodo **Polygauss** sul poligono non convesso, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	3.1e-04	9.1e-04	7.3e-05
	lsqnonneg m. esatti	1.1e-01	1.8e-01	1.3e-02
	lsqnonneg m. QMC	2.4e-02	7.2e-02	1.9e-02
	AFP m. esatti	2.1e-01	3.0e-01	2.2e-01
	AFP m. QMC	2.1e-01	3.0e-01	2.2e-01
$n = 20$	QMC	7.1e-04	5.2e-04	4.7e-04
	lsqnonneg m. esatti	2.3e-02	4.7e-03	2.8e-02
	lsqnonneg m. QMC	4.5e-02	4.6e-03	1.5e-02
	AFP m. esatti	2.1e-02	3.6e-03	2.6e-02
	AFP m. QMC	2.1e-02	3.4e-03	2.6e-02
$n = 30$	QMC	7.1e-04	5.2e-04	4.7e-04
	lsqnonneg m. esatti	4.3e-03	7.4e-03	6.4e-04
	lsqnonneg m. QMC	2.0e-02	8.8e-03	1.7e-02
	AFP m. esatti	3.3e-03	7.9e-03	6.5e-03
	AFP m. QMC	3.1e-03	7.8e-03	6.3e-03

Tabella 5.15: Errori relativi degli integrali di f_1 sul poligono non convesso, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo Polygauss.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	4.4e-04	5.6e-04	1.4e-04
	lsqnonneg m. esatti	1.0e-04	1.1e-04	1.0e-05
	lsqnonneg m. QMC	3.3e-04	3.9e-04	3.3e-04
	AFP m. esatti	6.7e-05	7.6e-05	9.9e-05
	AFP m. QMC	2.5e-05	3.4e-05	5.7e-05
$n = 20$	QMC	4.4e-04	5.6e-04	1.4e-04
	lsqnonneg m. esatti	7.1e-06	1.0e-05	4.3e-06
	lsqnonneg m. QMC	4.2e-04	5.8e-04	1.6e-04
	AFP m. esatti	3.2e-06	2.3e-05	8.2e-06
	AFP m. QMC	4.0e-05	2.0e-05	5.1e-05
$n = 30$	QMC	4.4e-04	5.6e-04	1.4e-04
	lsqnonneg m. esatti	1.3e-06	3.5e-06	5.0e-07
	lsqnonneg m. QMC	4.4e-04	5.6e-04	1.5e-04
	AFP m. esatti	2.5e-06	1.8e-07	8.3e-06
	AFP m. QMC	4.6e-05	4.3e-05	3.4e-05

Tabella 5.16: Errori relativi degli integrali di f_2 sul poligono non convesso, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo Polygauss.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	1.3e-02	2.0e-02	1.8e-02
	lsqnonneg m. esatti	1.3e-01	1.4e+00	3.9e-01
	lsqnonneg m. QMC	2.1e-01	1.4e-01	8.3e-01
	AFP m. esatti	8.9e-01	9.7e-01	7.5e-01
	AFP m. QMC	8.9e-01	9.7e-01	7.4e-01
$n = 20$	QMC	1.0e-02	1.8e-02	7.8e-03
	lsqnonneg m. esatti	1.3e-01	3.1e-01	4.0e-01
	lsqnonneg m. QMC	6.0e-01	2.7e-01	3.9e-01
	AFP m. esatti	2.4e-01	3.9e-02	3.7e-01
	AFP m. QMC	2.4e-01	2.3e-02	3.7e-01
$n = 30$	QMC	1.0e-02	1.0e-02	7.8e-03
	lsqnonneg m. esatti	1.6e-02	9.5e-01	9.5e-02
	lsqnonneg m. QMC	5.3e-02	9.1e-01	3.5e-01
	AFP m. esatti	1.4e-01	8.2e-01	5.6e-01
	AFP m. QMC	1.6e-01	8.2e-01	5.8e-01

Tabella 5.17: Errori relativi degli integrali di f_3 sul poligono non convesso, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo Polygauss.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	5.6e-01	3.2e-01	4.7e-01
	lsqnonneg m. esatti	4.3e+01	5.2e+01	2.1e+00
	lsqnonneg m. QMC	5.9e+01	4.3e+01	2.4e+01
	AFP m. esatti	5.9e+01	2.2e+01	3.0e+00
	AFP m. QMC	5.9e+01	2.2e+01	3.0e+00
$n = 20$	QMC	1.2e+00	1.3e+00	1.3e+00
	lsqnonneg m. esatti	4.1e+00	3.6e+00	8.0e+00
	lsqnonneg m. QMC	1.8e+01	1.6e+01	1.2e+01
	AFP m. esatti	1.3e+01	8.6e+00	1.2e+01
	AFP m. QMC	1.3e+01	8.6e+00	1.2e+01
$n = 30$	QMC	3.1e-01	1.0e+00	5.8e-01
	lsqnonneg m. esatti	2.7e+01	2.0e+01	2.9e-01
	lsqnonneg m. QMC	5.3e+01	1.1e+02	8.3e+01
	AFP m. esatti	5.4e+01	5.8e+01	5.2e+01
	AFP m. QMC	5.1e+01	5.7e+01	5.3e+01

Tabella 5.18: Errori relativi degli integrali di f_4 sul poligono non convesso, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo Polygauss.

5.3 Lente

Consideriamo la lente data dall'intersezione dei cerchi di centro e raggio rispettivamente $C_1 = (0, 0)$, $r_1 = 5$ e $C_2 = (4, 0)$, $r_2 = 3$. Le griglie iniziali utilizzate sono composte da punti di Halton. I momenti esatti sono stati calcolati usando nodi e pesi prodotti da **gqlens**, mentre i momenti QMC sono stati determinati a partire da una griglia iniziale di 600000 punti di Halton.

Nella Figura 5.3 sono riprodotti i nodi determinati da **gqlens** e da **lsqnonneg**, e gli AFP. Si può notare che questi ultimi due insiemi di punti sono distribuiti omogeneamente su tutto il dominio, accumulandosi verso il bordo.

La Tabella 5.19 riporta il numero di nodi usati dai vari metodi, al variare del grado n e della cardinalità N della griglia di Halton iniziale. A differenza dei poligoni precedenti, il numero dei nodi prodotti da **gqlens** è di poco maggiore a quello degli AFP. Il numero dei nodi determinati da **lsqnonneg** risulta anche qui quasi sempre uguale al numero degli AFP, tranne che per i nodi ricavati con momenti esatti, che sono di meno ai gradi $n = 20$ e, soprattutto, $n = 30$. I rapporti dei pesi (5.14) sono ottimali per **gqlens**, e molto buoni per i pesi degli AFP.

Nella Tabella 5.20 si possono vedere i risultati della cubatura ottenuta da **gqlens**. Gli integrali calcolati con le funzioni integrande f_3 ed f_4 mostrano differenze importanti al variare del grado n , in particolare f_4 . Nel caso di f_3 , tuttavia, si intuisce che l'integrale vero debba essere particolarmente piccolo (i risultati di **gqlens** sono dell'ordine di 10^{-16} e 10^{-21}). Poiché per f_3 e f_4 i

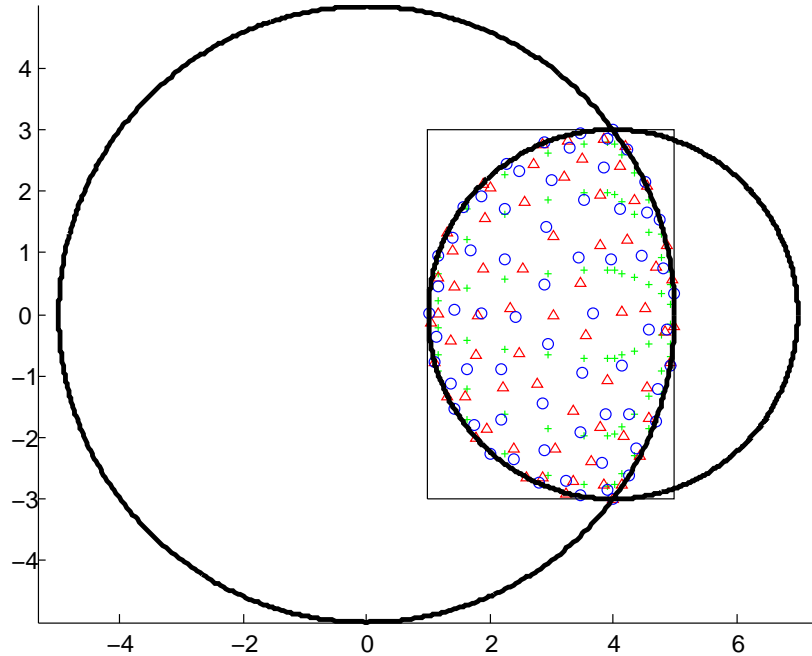


Figura 5.3: Lente con $N = 200000$, $n = 10$: distribuzione dei punti prodotti dal metodo **gqlens** (+), dei punti selezionati dalla griglia di Halton tramite **lsqnonneg** con momenti esatti (Δ), e degli AFP (\circ).

risultati di **gqlens** sono poco precisi, ci si attende che i rispettivi errori relativi commessi dai vari metodi possano essere grandi.

Nelle Tabelle 5.21-5.24 sono presentati gli errori relativi dei vari metodi rispetto ai risultati di **gqlens** (Tabella 5.20). Gli errori sono particolarmente grandi per le funzioni f_3 e f_4 , come già preannunciato. Per le funzioni f_1 e f_2 , invece, gli errori sono piccoli, il che come in precedenza è in parte da attribuire al Teorema di Koksma-Hlawka.

Le seguenti osservazioni riguardo agli errori commessi dai vari metodi riguardano in particolare le funzioni test f_1 ed f_2 , dal momento che, per il motivo visto sopra, f_3 e f_4 hanno errori relativi di grande entità e con un andamento imprevedibile. L'integrazione QMC migliora leggermente all'aumentare della cardinalità N della griglia di Halton iniziale. Gli errori dati dal metodo di compressione **lsqnonneg** e dalla cubatura su AFP con momenti esatti calano al crescere del grado n , specialmente per f_2 . Entrambi questi metodi con i momenti QMC, invece, migliorano sia all'aumentare di n sia all'aumentare di N , ma molto lentamente per f_1 e quasi per nulla nel caso di f_2 .

Si può infine osservare che per la funzione test f_1 i vari metodi risultano in generale equivalenti. Nel caso di f_2 , invece, i metodi di cubatura migliori sono di gran lunga la compressione via **lsqnonneg** e l'integrazione sugli AFP con momenti esatti. Inoltre, **lsqnonneg** e AFP con momenti QMC danno risultati simili a quelli dell'integrazione QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	gqlens	72 (1.00)	72 (1.00)	72 (1.00)
	QMC	37968	75925	151880
	lsqnonneg m. esatti	66	66	66
	lsqnonneg m. QMC	66	66	66
	AFP m. esatti	66 (1.02)	66 (1.00)	66 (1.04)
	AFP m. QMC	66 (1.02)	66 (1.00)	66 (1.04)
$n = 20$	gqlens	242 (1.00)	242 (1.00)	242 (1.00)
	QMC	37968	75925	151880
	lsqnonneg m. esatti	210	209	207
	lsqnonneg m. QMC	231	231	231
	AFP m. esatti	231 (1.03)	231 (1.04)	231 (1.04)
	AFP m. QMC	231 (1.03)	231 (1.04)	231 (1.04)
$n = 30$	gqlens	512 (1.00)	512 (1.00)	512 (1.00)
	QMC	37968	75925	151880
	lsqnonneg m. esatti	416	413	409
	lsqnonneg m. QMC	496	495	495
	AFP m. esatti	496 (1.01)	496 (1.03)	496 (1.01)
	AFP m. QMC	496 (1.02)	496 (1.04)	496 (1.02)

Tabella 5.19: Numero dei nodi sulla lente prodotti da **gqlens**, dei punti quasi-random contenuti nel dominio, dei punti estratti da **lsqnonneg** con momenti esatti/QMC, e degli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / \sum w_i$.

$n = 10$	5.1930717939878662e-02	$n = 10$	5.5084225946229793e+01
f_1 $n = 20$	5.3712904940154287e-02	f_2 $n = 20$	5.5084752353784189e+01
$n = 30$	5.3362997480936369e-02	$n = 30$	5.5084763113715269e+01
$n = 10$	1.1886542819142382e-21	$n = 10$	-5.2700435704784931e-01
f_3 $n = 20$	7.4109650472942533e-16	f_4 $n = 20$	5.2189935464641790e-02
$n = 30$	6.4147923897608779e-16	$n = 30$	1.5742704610932448e+00

Tabella 5.20: Valori degli integrali ottenuti dal metodo **gqlens** sulla lente, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	2.0e-02	2.2e-02	2.7e-02
	lsqnonneg m. esatti	2.9e-02	1.7e-02	4.4e-02
	lsqnonneg m. QMC	3.9e-02	2.9e-02	2.6e-02
	AFP m. esatti	1.8e-02	2.4e-02	5.7e-02
	AFP m. QMC	1.8e-02	2.4e-02	5.7e-02
$n = 20$	QMC	1.4e-02	1.2e-02	7.2e-03
	lsqnonneg m. esatti	8.5e-04	1.8e-02	2.3e-02
	lsqnonneg m. QMC	1.1e-02	1.4e-04	1.3e-02
	AFP m. esatti	6.0e-02	2.4e-02	1.7e-02
	AFP m. QMC	6.0e-02	2.4e-02	1.7e-02
$n = 30$	QMC	7.6e-03	5.7e-03	6.5e-04
	lsqnonneg m. esatti	3.3e-03	1.5e-03	9.3e-04
	lsqnonneg m. QMC	1.9e-02	7.6e-03	9.1e-04
	AFP m. esatti	1.6e-03	2.1e-03	5.5e-03
	AFP m. QMC	1.4e-03	4.0e-03	4.5e-04

Tabella 5.21: Errori relativi degli integrali di f_1 sulla lente, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `gqlens`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	2.6e-04	7.6e-05	6.9e-05
	lsqnonneg m. esatti	7.4e-05	2.0e-05	6.9e-05
	lsqnonneg m. QMC	2.2e-04	6.4e-05	5.2e-05
	AFP m. esatti	3.0e-06	3.2e-05	7.6e-05
	AFP m. QMC	5.2e-05	8.1e-05	2.7e-05
$n = 20$	QMC	2.5e-04	8.6e-05	5.9e-05
	lsqnonneg m. esatti	4.0e-07	1.3e-07	1.0e-06
	lsqnonneg m. QMC	2.5e-04	8.7e-05	5.9e-05
	AFP m. esatti	1.0e-06	3.6e-07	1.1e-06
	AFP m. QMC	5.0e-05	4.8e-05	4.8e-05
$n = 30$	QMC	2.5e-04	8.6e-05	5.9e-05
	lsqnonneg m. esatti	2.9e-07	7.6e-08	3.5e-07
	lsqnonneg m. QMC	2.5e-04	8.6e-05	5.9e-05
	AFP m. esatti	1.1e-08	7.8e-09	4.7e-08
	AFP m. QMC	4.9e-05	4.9e-05	4.9e-05

Tabella 5.22: Errori relativi degli integrali di f_2 sulla lente, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `gqlens`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	4.2e+05	3.4e+05	4.2e+05
	lsqnonneg m. esatti	1.9e-01	7.3e+04	3.5e+00
	lsqnonneg m. QMC	8.2e+03	4.2e+03	1.4e+04
	AFP m. esatti	2.4e+03	9.2e-01	9.0e+01
	AFP m. QMC	2.3e+03	9.2e-01	8.9e+01
$n = 20$	QMC	3.3e-01	4.5e-01	3.3e-01
	lsqnonneg m. esatti	1.0e+00	9.3e-01	2.7e-01
	lsqnonneg m. QMC	1.0e+00	3.4e-01	9.9e-01
	AFP m. esatti	6.3e+00	8.4e+00	3.2e+00
	AFP m. QMC	6.4e+00	8.3e+00	3.1e+00
$n = 30$	QMC	2.2e-01	3.6e-01	2.3e-01
	lsqnonneg m. esatti	5.3e-01	1.2e-01	6.6e-01
	lsqnonneg m. QMC	7.6e-01	2.7e-02	1.7e-01
	AFP m. esatti	5.0e-01	1.1e+00	1.4e+00
	AFP m. QMC	1.8e+00	1.0e+00	6.9e-01

Tabella 5.23: Errori relativi degli integrali di f_3 sulla lente, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `gqlens`.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	8.8e-01	1.0e+00	9.9e-01
	lsqnonneg m. esatti	2.0e+00	1.4e+00	4.4e+00
	lsqnonneg m. QMC	1.8e+00	5.5e+00	2.6e+00
	AFP m. esatti	3.1e+00	4.2e+00	2.1e+00
	AFP m. QMC	3.1e+00	4.2e+00	2.1e+00
$n = 20$	QMC	2.2e+00	8.8e-01	1.1e+00
	lsqnonneg m. esatti	1.1e+01	1.2e+01	1.7e+01
	lsqnonneg m. QMC	3.8e+01	1.5e+01	1.7e+01
	AFP m. esatti	2.6e+01	4.5e+00	1.8e+01
	AFP m. QMC	2.6e+01	4.5e+00	1.8e+01
$n = 30$	QMC	1.0e+00	1.0e+00	1.0e+00
	lsqnonneg m. esatti	4.1e-01	1.1e+00	1.2e+00
	lsqnonneg m. QMC	7.2e-01	1.7e+00	4.1e-01
	AFP m. esatti	1.3e+00	7.4e-01	4.0e-01
	AFP m. QMC	1.2e+00	4.4e-01	4.2e-01

Tabella 5.24: Errori relativi degli integrali di f_4 sulla lente, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo `gqlens`.

5.4 Dominio composito

Il dominio composito che prendiamo in esame è dato dall'unione di:

- un cerchio di centro $C = (0, 0)$ e raggio $r = 3$;
- un quadrato di vertici $P_1 = (0, 0)$, $P_2 = (4, 0)$, $P_3 = (4, 4)$, $P_4 = (0, 4)$;
- un poligono non convesso di vertici $P_1 = (1, 1)$, $P_2 = (6, 2)$, $P_3 = (7, 4)$, $P_4 = (10, 3)$, $P_5 = (9, 6)$, $P_6 = (6, 7)$, $P_7 = (4, 5)$, $P_8 = (1, 6)$, $P_9 = (1, 1)$.

Per questo tipo di dominio non disponiamo di alcun metodo di cubatura con grado algebrico esatto né sono noti i momenti esatti, per cui i metodi che metteremo a confronto sono soltanto il metodo d'integrazione QMC, la compressione via `lsqnonneg` e la cubatura sugli AFP con momenti approssimati QMC. Il metodo che prendiamo come riferimento per gli errori relativi è quello QMC, e le griglie quasi-random iniziali sono formate da punti di Halton. I momenti QMC sono stati determinati a partire da una griglia iniziale di 600000 punti di Halton.

Nella Figura 5.4 sono raffigurati i nodi prodotti da `lsqnonneg` e gli AFP; è possibile osservare come le distribuzioni di questi punti siano omogenee su tutto il dominio, con addensamenti verso il bordo.

Nella Tabella 5.25 è riportato il numero di nodi usati da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Il numero dei nodi determinati da `lsqnonneg` risulta uguale a quello degli AFP. I rapporti dei pesi (5.14) degli AFP, inoltre, sono abbastanza buoni al grado

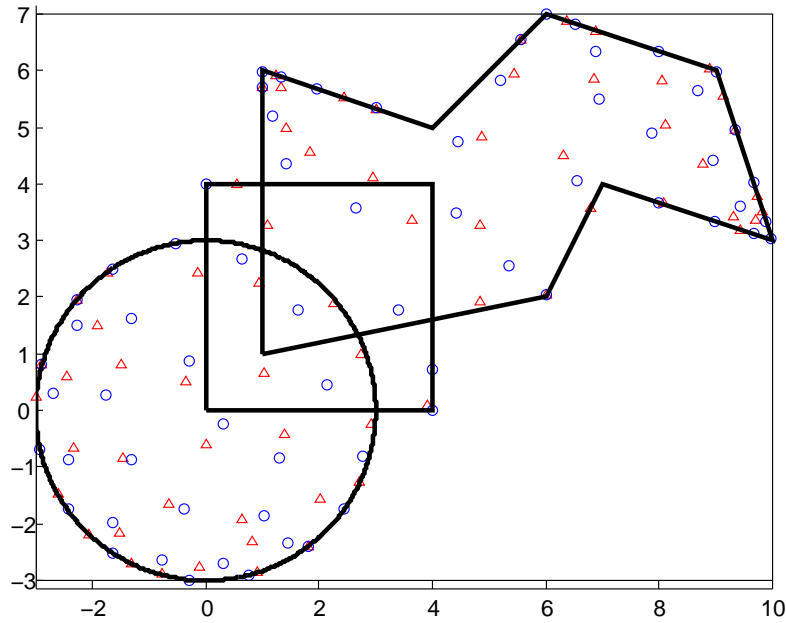


Figura 5.4: Dominio composito con $N = 200000$, $n = 10$: distribuzione dei punti selezionati dalla griglia di Halton tramite `lsqnonneg` con momenti QMC (\triangle), e degli AFP (\circ).

$n = 10$, ancora accettabili al grado $n = 20$ ma molto brutti per $n = 30$. Per il grado $n = 30$, quindi, può accadere che la cubatura sugli AFP non dia risultati buoni.

Nella Tabella 5.26 sono elencati i risultati della cubatura ottenuta dal metodo QMC. Come si può vedere, gli integrali calcolati mostrano leggere variazioni al variare di N , in particolare per f_3 e, soprattutto, f_4 . Anche in questo caso, dunque, si prevede che per le funzioni f_3 ed f_4 gli errori relativi dei vari metodi rispetto a QMC possano essere grandi, essendo i risultati stessi di QMC non precisi.

Riportiamo nelle Tabelle 5.27-5.30 gli errori relativi commessi da `lsqnonneg` e dagli AFP rispetto ai risultati dell'integrazione QMC (Tabella 5.26). Come già anticipato, gli errori per le funzioni f_3 e f_4 sono molto grandi. Inoltre, salta immediatamente all'occhio il fatto che la cubatura sugli AFP al grado $n = 30$ sia quasi sempre disastrosa, e il motivo risiede nel rapporto (5.14) dei pesi degli AFP, ovvero nell'entità dei pesi negativi rispetto a quelli positivi. Con gli altri gradi, il metodo `lsqnonneg` risulta in generale equivalente o migliore di quello degli AFP per f_1 , mentre nel caso di f_2 gli AFP sono più efficienti per gradi bassi ($n = 10$) ed equivalenti a `lsqnonneg` attorno al grado $n = 20$. Gli errori dei risultati ottenuti da `lsqnonneg`, infine, migliorano sensibilmente all'aumentare del grado n , al contrario di quelli prodotti con gli AFP.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	QMC	23323	46619	93269
	lsqnonneg	66	66	66
	AFP	66 (1.03)	66 (1.08)	66 (1.03)
$n = 20$	QMC	23323	46619	93269
	lsqnonneg	231	231	231
	AFP	231 (1.35)	231 (1.28)	231 (1.29)
$n = 30$	QMC	23323	46619	93269
	lsqnonneg	496	496	496
	AFP	496 (8544.15)	496 (5092.41)	496 (20346.70)

Tabella 5.25: Numero dei punti quasi-random contenuti nel dominio composito, dei punti estratti da **lsqnonneg**, e degli AFP, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / |\sum w_i|$.

f_1	$N = 50000$	1.4937237186766762e+01
	$N = 100000$	1.4907683736673341e+01
	$N = 200000$	1.4895939195211284e+01
f_2	$N = 50000$	2.4631282197075410e+02
	$N = 100000$	2.4606364047901192e+02
	$N = 200000$	2.4621007671000882e+02
f_3	$N = 50000$	3.3809966978309541e-02
	$N = 100000$	3.2280929902177180e-02
	$N = 200000$	3.1358389918444494e-02
f_4	$N = 50000$	3.6782925584934489e-01
	$N = 100000$	1.9089015913588764e-01
	$N = 200000$	1.3990115207157169e-01

Tabella 5.26: Valori degli integrali sul dominio composito ottenuti dal metodo QMC usando punti di Halton, al variare della funzione integranda e della cardinalità N della griglia iniziale sul rettangolo circoscritto al dominio.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	lsqnonneg	4.1e-02	3.4e-02	5.3e-02
	AFP	5.7e-02	1.1e-01	3.0e-02
$n = 20$	lsqnonneg	1.3e-02	4.1e-03	6.9e-03
	AFP	3.0e-02	4.3e-03	1.1e-02
$n = 30$	lsqnonneg	3.6e-03	2.8e-03	3.8e-03
	AFP	2.8e+01	9.7e-02	1.4e+01

Tabella 5.27: Errori relativi degli integrali di f_1 sul dominio composito, ottenuti da **lsqnonneg** e dagli AFP, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	lsqnonneg	3.1e-03	1.5e-03	1.6e-03
	AFP	1.7e-04	3.1e-05	4.4e-04
$n = 20$	lsqnonneg	3.9e-04	1.7e-04	3.0e-04
	AFP	9.2e-04	3.9e-04	2.6e-04
$n = 30$	lsqnonneg	6.3e-05	2.4e-07	1.7e-04
	AFP	1.2e-01	9.7e-02	3.7e+00

Tabella 5.28: Errori relativi degli integrali di f_2 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	lsqnonneg	1.0e+00	1.0e+00	1.0e+00
	AFP	1.0e+00	1.0e+00	1.0e+00
$n = 20$	lsqnonneg	1.0e+00	7.8e-01	9.6e-01
	AFP	8.2e-01	5.4e-01	1.0e+00
$n = 30$	lsqnonneg	1.0e+00	9.5e-01	4.8e+00
	AFP	5.7e+00	6.5e+02	3.7e+04

Tabella 5.29: Errori relativi degli integrali di f_3 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 10$	lsqnonneg	3.6e+00	1.5e+01	3.3e+00
	AFP	9.5e+00	2.4e+01	8.2e+01
$n = 20$	lsqnonneg	2.1e+01	6.6e+00	6.0e+00
	AFP	3.9e+00	2.0e+01	2.4e+01
$n = 30$	lsqnonneg	3.4e+00	2.3e+01	2.6e+01
	AFP	1.5e+04	6.1e+04	3.7e+05

Tabella 5.30: Errori relativi degli integrali di f_4 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati del metodo QMC.

Capitolo 6

Esperimenti numerici su domini 3D

In questo capitolo faremo alcuni esperimenti numerici sul cubo, sul cono, sulla piramide e su un dominio composito 3D, mettendo a confronto il metodo di integrazione quasi-Monte Carlo (d'ora in poi "QMC"), la compressione del numero dei nodi quasi-random tramite `lsqnonneg`, e quella sui punti approssimati di Fekete estratti da un insieme di punti quasi-random (d'ora in poi "AFP Estratti") con alcuni metodi di cubatura già noti che prenderemo come riferimento.

La cubatura con il metodo QMC viene eseguita utilizzando la formula (5.1). Per brevità, come punti quasi-random si considerano solo nodi di Halton. L'insieme di punti di Halton viene ricavato da una griglia di Halton iniziale, con una certa cardinalità N , creata su un parallelepipedo rettangolo con i lati allineati agli assi coordinati e tale che contenga il dominio. Dopodiché, tramite funzioni costruite ad hoc, vengono selezionati solo i punti che si trovano all'interno del dominio, che saranno in numero $N_{qmc} \leq N$, e a questi viene applicata la formula (5.1). La griglia iniziale viene creata un'unica volta all'inizio, mediante i comandi

```
P = haltonset(3);  
Pmc = net(P,N);
```

dove N è la cardinalità dell'insieme. In questo modo, i confronti tra i metodi QMC e di compressione mediante `lsqnonneg` e AFP Estratti vengono condotti sul medesimo insieme di punti.

Anche nel caso trivariato il metodo di compressione dei punti quasi-random via `lsqnonneg` è implementato dalla funzione Matlab `compresscub` (vedasi CD allegato al lavoro), che funziona analogamente al caso 2-dimensionale sia per produrre i momenti da passare in input a `lsqnonneg`, sia per determinare i nuovi pesi e nodi di cubatura. La stima (5.9) sull'errore di cubatura data dal Teorema (14) vale anche nel caso trivariato.

Gli AFP Estratti vengono scelti dall'insieme di punti di Halton tramite la routine di Matlab `approxfek` [56], che implementa l'Algoritmo AFP Greedy descritto nel Capitolo 4. Anche in questo caso, se si dispone dei momenti esatti sul dominio, si calcolano i pesi esatti per i nodi AFP Estratti, altrimenti si possono calcolare i momenti QMC in maniera analoga a quanto fatto in `compresscube`,

ottenendo pesi approssimati per i nodi AFP Estratti. In entrambi i casi, i pesi calcolati possono essere in generale negativi.

Il metodo di cubatura che prendiamo come riferimento per il cubo, è quello dato dalla formula di Gauss-Chebyshev prodotto tensoriale, che ha grado di esattezza polinomiale. Per il cono e per la piramide, invece, come base di confronto è stata scelta l'integrazione sui punti approssimati di Fekete calcolati dal pacchetto **3dWAM** [33, 34]. A differenza degli AFP Estratti, si tratta dei punti approssimati di Fekete estratti da una WAM sul cono o sulla piramide (d'ora in poi li chiameremo sinteticamente "AFP da WAM" per distinguerli dagli AFP Estratti). Per il dominio composito, infine, terremo come metodo di riferimento per il calcolo degli errori il metodo di integrazione QMC.

Un'osservazione importante da fare, è che gli integrali che vengono presi come valori di riferimento sono comunque approssimati. Non solo, ma il confronto di ciascun metodo rispetto ai risultati del metodo di riferimento viene condotto *con i medesimi n e N* . Nel caso del cubo, per ogni grado n , si è preso come integrale di riferimento quello calcolato da Gauss-Chebyshev prodotto tensoriale proprio al grado n : non si è usato Gauss-Chebyshev con un grado più elevato $\hat{n} \gg n$ per avere un risultato di cubatura più vicino a quello esatto. Analogamente con **3dWAM** per il cono e la piramide. Per il dominio composito vale un discorso simile: si è scelta l'integrazione QMC come metodo di riferimento, e la si è usata per produrre integrali con lo stesso valore N degli altri metodi, non con un valore superiore $\hat{N} \gg N$. Ovvero, l'analisi degli errori è stata condotta rispetto al (comportamento del) metodo preso come riferimento, non rispetto all'integrale esatto vero e proprio.

Nelle prossime sezioni presenteremo i risultati degli esperimenti numerici effettuati. Le funzioni test che prendiamo in considerazione sono

$$\begin{aligned} f_1(x, y, z) &= 0.75e^{-\frac{1}{4}((9x-2)^2+(9y-2)^2+(9z-2)^2)} \\ &\quad + 0.75e^{-\frac{1}{49}(9x+1)^2-\frac{1}{10}(9y+1)-\frac{1}{10}(9z+1)} \\ &\quad + 0.5e^{-\frac{1}{4}((9x-7)^2+(9y-3)^2+(9z-5)^2)} \\ &\quad - 0.2e^{-(9x-4)^2-(9y-7)^2-(9z-5)^2} \end{aligned} \quad (6.1)$$

$$f_2(x, y, z) = \sqrt{(x-0.4)^2 + (x-0.4)^2 + (x-0.4)^2} \quad (6.2)$$

$$f_3(x, y, z) = \cos(4(x+y+z)) \quad (6.3)$$

$$f_4(x, y, z) = \frac{1}{1+16(x^2+y^2+z^2)} \quad (6.4)$$

$$f_5(x, y, z) = \sqrt{(x^2+y^2+z^2)^3}. \quad (6.5)$$

Per ogni funzione test vengono condotti esperimenti su tre differenti gradi polinomiali, $n = 5, 7, 9$, e su tre griglie iniziali di punti di Halton, rispettivamente di cardinalità $N = 50000, 100000, 200000$. La scelta molto bassa dei gradi è dovuta alla complessità numerica dei problemi, che oltre ad avere tempi di esecuzione molto lunghi, porta rapidamente ad errori di tipo "out of memory". La funzione Matlab **approxfek**, in particolare, richiede tempi di calcolo estremamente lunghi; tuttavia, allo stesso modo di **lsqnonneg**, per ogni dato dominio e grado n , questa routine ne determina una volta per tutte i nodi e i pesi, che quindi non devono essere più ricalcolati.

Ricordiamo che i metodi presi in considerazione dipendono in maniera diversa dal grado n e dalla cardinalità della griglia di partenza N , nello specifico:

$n = 5$	$n = 7$	$n = 9$
1.4201e-05	2.0509e-05	3.7675e-05

Tabella 6.1: Errori relativi dei momenti QMC rispetto ai momenti esatti sul cubo.

- il metodo di Gauss-Chebyshev prodotto tensoriale dipende solamente da n , e il numero dei nodi prodotti è n^3 ;
- la cubatura mediante **3dWAM** dipende solo da n , e gli AFP da WAM sono in numero $\eta_n = \dim(\mathbb{P}_n^3) = (n+1)(n+2)(n+3)/6$;
- l'integrazione QMC dipende soltanto da N ;
- il metodo di compressione tramite **lsqnonneg** dipende sia da n sia da N ; in particolare, sia il valore degli integrali sia il numero dei nodi estratti da **lsqnonneg** dipendono da n e N ;
- il calcolo dei nodi AFP Estratti dipende sia da n sia da N . Tuttavia, mentre gli integrali calcolati sugli AFP Estratti sono determinati sia da n sia da N , il numero dei nodi dipende soltanto da n , essendo uguale alla dimensione del spazio polinomiale \mathbb{P}_n^3 , ovvero $\eta_n = \dim(\mathbb{P}_n^3) = (n+1)(n+2)(n+3)/6$.

Per ogni dominio verrà riportata la tabella con il numero di nodi impiegati da ciascun metodo, al variare del grado n e dal numero di punti N della griglia quasi-random iniziale. Nel caso di pesi potenzialmente negativi, viene calcolato il rapporto

$$\rho = \frac{\sum_i |w_i|}{|\sum_i w_i|} \quad (6.6)$$

che assume valori nell'intervallo $[1, +\infty[$ e che, come spiegato nel capitolo precedente, è un parametro di stabilità dell'algoritmo.

Non essendo noti i momenti esatti né per il cono né per la piramide né per il dominio composito scelto, useremo quelli QMC per determinare i pesi di cubatura di **lsqnonneg**, degli AFP Estratti e - nel caso del cono e della piramide - degli AFP da WAM. Quanto al cubo, invece, sebbene disponiamo dei momenti esatti, effettuiamo ugualmente la cubatura via **lsqnonneg** e sugli AFP Estratti sia con momenti esatti sia con momenti QMC. Confrontando i risultati, infatti, è possibile stabilire quanto possa essere ragionevole - in termini di buona approssimazione - usare i momenti QMC in mancanza di momenti esatti (ossia in tutti gli altri domini). Nella Tabella 6.1 sono riportati gli errori relativi dei momenti QMC rispetto ai momenti esatti per il cubo, arrotondati alla quarta cifra decimale; come si può vedere, sono abbastanza piccoli (dell'ordine di 10^{-5}), anche se tendono molto lentamente a crescere con il grado n .

Tutti gli esperimenti numerici sono stati realizzati con Matlab 7.10.0, su un 3.00 GHz Intel Core 2 Duo con 4 GB di RAM. I tempi di esecuzione dei metodi di Gauss-Chebyshev prodotto tensoriale, di **3dWAM** e di QMC sono inferiori al secondo; quelli di **lsqnonneg** vanno da 1 secondo ($n = 5$, $N = 50000$) a 1 minuto ($n = 9$, $N = 200000$); quelli di **approxfek** per il calcolo degli AFP Estratti vanno da alcuni secondi ($n = 5$, $N = 50000$) a 45 minuti ($n = 9$, $N = 200000$).

6.1 Cubo

Il primo dominio che prendiamo in considerazione è il cubo unitario $[0, 1]^3$. Le griglie quasi-random iniziali sono costituite da punti di Halton. I momenti esatti sono stati calcolati usando nodi e pesi prodotti dal metodo Gauss-Chebyshev prodotto tensoriale, mentre i momenti QMC sono stati determinati a partire da una griglia iniziale di 1 milione di punti di Halton.

Nella Figura 6.1 sono raffigurati i nodi prodotti da Gauss-Chebyshev prodotto tensoriale e da `lsqnonneg`, e gli AFP Estratti. Sia i punti determinati da `lsqnonneg` sia gli AFP Estratti hanno distribuzione omogenea su tutto il dominio, accumulandosi verso il bordo.

Nella Tabella 6.2 è riportato il numero di nodi usati da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton. Il numero di nodi impiegati da Gauss-Chebyshev prodotto tensoriale è n^3 , mentre gli AFP Estratti sono $(n+1)(n+2)(n+3)/6$. I punti determinati da `lsqnonneg` sono generalmente in ugual numero (o di poco inferiore, per i valori di N maggiori) a quello degli AFP Estratti. I rapporti dei pesi si dimostrano ottimali per Gauss-Chebyshev prodotto tensoriale; quelli degli AFP Estratti, invece, registrano una presenza non irrilevante di pesi negativi, essendo ρ compreso tra 1.41 e 2.42. Dalla teoria ci si attende, quindi, che i risultati ottenuti dagli AFP Estratti possano essere soggetti a fenomeni di instabilità, in particolare per $n = 5$ dove i rapporti ρ sono più elevati.

In Tabella 6.3 sono elencati i risultati della cubatura ottenuta da Gauss-

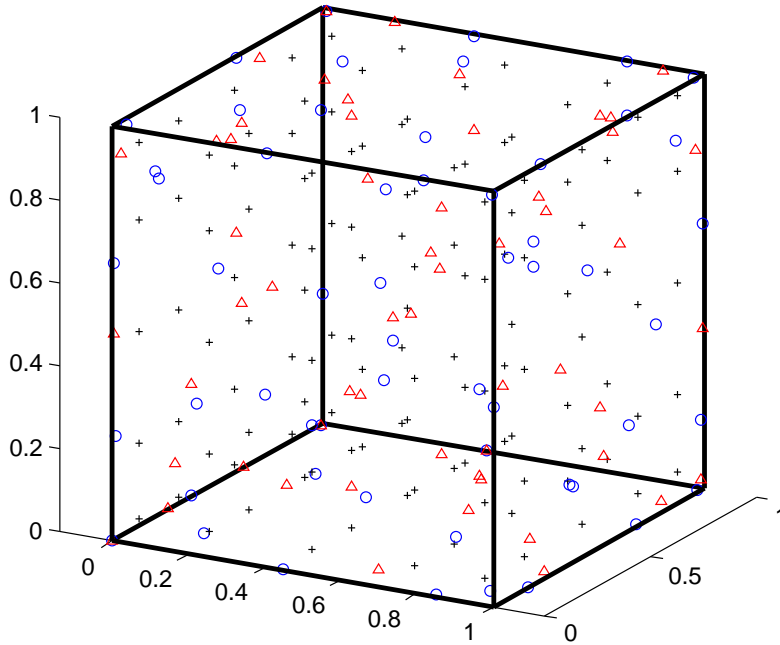


Figura 6.1: Cubo con $N = 100000$, $n = 5$: distribuzione dei punti prodotti da Gauss-Chebyshev prodotto tensoriale (+), dei punti estratti dalla griglia di Halton tramite `lsqnonneg` con momenti esatti (Δ), e degli AFP (o).

Chebyshev prodotto tensoriale, rispetto ai quali nelle Tabelle 6.4-6.8 sono stati calcolati gli errori relativi compiuti (a pari grado n) dall'integrazione QMC, dalla compressione via `lsqnonneg` con momenti esatti e con momenti QMC, e dalla cubatura condotta sugli AFP Estratti con momenti esatti e con momenti QMC.

La funzione f_2 ha variazione piccola sul cubo $[0, 1]^3$, per cui in virtù del Teorema di Koksma-Hlawka (Teorema 5) ci aspetteremmo risultati buoni per l'integrazione QMC; gli errori di QMC per f_2 , invece, sono maggiori rispetto a quelli ottenuti per le altre funzioni test. Il motivo più probabile è che i valori integrali di riferimento per la funzione f_2 (ossia quelli in Tabella 6.3) sono meno precisi rispetto a quelli delle altre funzioni, e quindi necessariamente anche gli errori calcolati rispetto ad essi risultano più rozzi. Per tutte le altre funzioni test, in particolare f_3 , f_4 e f_5 , il metodo QMC funziona molto bene, pur non registrando significativi miglioramenti all'aumentare della cardinalità N della griglia di Halton.

La cubatura con `lsqnonneg` e sugli AFP Estratti risulta buona, anche se occorre salire di grado affinché diventi migliore di QMC. Da questo punto di vista, il vantaggio di QMC è proprio quello di non dipendere dal grado, il che si rivela utile in dimensione 3, perché i tempi e la complessità computazionale del metodo `lsqnonneg` crescono in fretta, e quelli dell'estrazione degli AFP impediscono la cubatura già per gradi superiori (e talvolta uguali) a 9.

In generale, per i gradi considerati, il metodo `lsqnonneg` e quello degli AFP Estratti producono risultati equivalenti. Solo per la funzione f_3 (e un po' anche per f_4) si notano errori più cospicui da parte degli AFP Estratti: questo fenomeno è probabilmente dovuto alla elevata variazione della funzione test sul dominio, combinata con l'effetto dei pesi negativi.

Si noti, infine, che l'utilizzo dei momenti esatti e di quelli QMC per i metodi `lsqnonneg` e per gli AFP danno risultati equivalenti. C'è da dire, tuttavia, che si stanno considerando gradi piuttosto bassi, e, poiché i momenti esatti vengono calcolati dal metodo Gauss-Chebyshev prodotto tensoriale che ha grado di esattezza polinomiale, è da supporre che i momenti esatti dimostrino una migliore efficienza per gradi più alti.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	GC	125 (1.00)	125 (1.00)	125 (1.00)
	lsqnonneg m. esatti	56	56	56
	lsqnonneg m. QMC	56	56	56
	AFP m. esatti	56 (2.42)	56 (1.66)	56 (1.96)
	AFP m. QMC	56 (2.42)	56 (1.66)	56 (1.96)
$n = 7$	GC	343 (1.00)	343 (1.00)	343 (1.00)
	lsqnonneg m. esatti	120	120	120
	lsqnonneg m. QMC	120	119	119
	AFP m. esatti	120 (1.87)	120 (1.51)	120 (1.78)
	AFP m. QMC	120 (1.87)	120 (1.51)	120 (1.78)
$n = 9$	GC	729 (1.00)	729 (1.00)	729 (1.00)
	lsqnonneg m. esatti	220	220	220
	lsqnonneg m. QMC	220	119	216
	AFP m. esatti	220 (1.41)	220 (1.43)	220 (1.48)
	AFP m. QMC	220 (1.41)	220 (1.43)	220 (1.48)

Tabella 6.2: Numero dei punti sul cubo prodotti da Gauss-Chebyshev prodotto tensoriale, dei punti estratti da `lsqnonneg` con momenti esatti/QMC, e degli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / |\sum w_i|$.

f_1	$n = 5$	2.0730417432515380e-01
	$n = 7$	2.0855399019154303e-01
	$n = 9$	2.0854461786826145e-01
f_2	$n = 5$	4.5839946332005521e-01
	$n = 7$	4.5645191284376058e-01
	$n = 9$	4.5368370106574341e-01
f_3	$n = 5$	9.0235463330332180e-02
	$n = 7$	9.0235236631536708e-02
	$n = 9$	9.0235236621786896e-02
f_4	$n = 5$	8.6353934772196730e-02
	$n = 7$	8.6381765269192778e-02
	$n = 9$	8.6380650567305553e-02
f_5	$n = 5$	1.1013957270366264e+00
	$n = 7$	1.1013957098668690e+00
	$n = 9$	1.1013957094255211e+00

Tabella 6.3: Valori degli integrali ottenuti da Gauss-Chebyshev prodotto tensoriale sul cubo, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	5.96e-03	5.82e-03	5.79e-03
	lsqnonneg m. esatti	7.71e-02	2.83e-02	3.27e-02
	lsqnonneg m. QMC	1.37e-01	6.78e-02	3.40e-02
	AFP m. esatti	3.87e-02	1.40e-01	4.66e-02
	AFP m. QMC	3.87e-02	1.40e-01	4.66e-02
$n = 7$	QMC	7.21e-05	2.12e-04	2.39e-04
	lsqnonneg m. esatti	2.29e-02	2.34e-02	2.72e-02
	lsqnonneg m. QMC	8.40e-03	3.15e-02	5.23e-02
	AFP m. esatti	1.89e-02	1.82e-02	7.75e-03
	AFP m. QMC	1.89e-02	1.82e-02	7.76e-03
$n = 9$	QMC	2.72e-05	1.67e-04	1.94e-04
	lsqnonneg m. esatti	1.88e-03	6.29e-04	4.61e-03
	lsqnonneg m. QMC	5.17e-03	1.35e-02	1.31e-03
	AFP m. esatti	2.10e-03	8.98e-04	1.93e-02
	AFP m. QMC	2.11e-03	8.95e-04	1.93e-02

Tabella 6.4: Errori relativi degli integrali di f_1 sul cubo, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/quasi-Monte Carlo, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di Gauss-Chebyshev prodotto tensoriale.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	1.76e-02	1.76e-02	1.76e-02
	lsqnonneg m. esatti	2.15e-02	2.12e-03	1.05e-02
	lsqnonneg m. QMC	3.52e-02	3.05e-03	1.18e-02
	AFP m. esatti	2.37e-02	3.01e-03	4.21e-02
	AFP m. QMC	2.37e-02	3.01e-03	4.21e-02
$n = 7$	QMC	1.34e-02	1.34e-02	1.34e-02
	lsqnonneg m. esatti	6.33e-03	8.96e-03	1.17e-02
	lsqnonneg m. QMC	1.18e-02	1.35e-02	2.87e-02
	AFP m. esatti	3.06e-02	1.87e-02	1.42e-02
	AFP m. QMC	3.06e-02	1.87e-02	1.42e-02
$n = 9$	QMC	7.42e-03	7.40e-03	7.39e-03
	lsqnonneg m. esatti	1.05e-02	1.66e-03	7.92e-03
	lsqnonneg m. QMC	6.48e-03	5.57e-03	6.29e-03
	AFP m. esatti	8.42e-03	3.70e-03	4.36e-03
	AFP m. QMC	8.42e-03	3.70e-03	4.36e-03

Tabella 6.5: Errori relativi degli integrali di f_2 sul cubo, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di Gauss-Chebyshev prodotto tensoriale.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	1.46e-04	1.15e-05	1.85e-04
	lsqnonneg m. esatti	6.29e-03	2.09e-01	3.05e-01
	lsqnonneg m. QMC	2.09e-01	4.98e-02	2.75e-02
	AFP m. esatti	4.91e+00	4.36e-01	1.10e+00
	AFP m. QMC	4.91e+00	4.36e-01	1.10e+00
$n = 7$	QMC	1.43e-04	8.94e-06	1.82e-04
	lsqnonneg m. esatti	1.07e-01	8.78e-02	1.07e-01
	lsqnonneg m. QMC	1.05e-01	8.30e-02	9.10e-02
	AFP m. esatti	1.32e-01	1.28e-01	1.00e-01
	AFP m. QMC	1.32e-01	1.28e-01	1.00e-01
$n = 9$	QMC	1.43e-04	8.94e-06	1.82e-04
	lsqnonneg m. esatti	4.53e-03	7.43e-04	3.30e-03
	lsqnonneg m. QMC	8.03e-03	4.67e-03	1.65e-04
	AFP m. esatti	1.13e-03	6.93e-03	1.49e-02
	AFP m. QMC	1.12e-03	6.94e-03	1.49e-02

Tabella 6.6: Errori relativi degli integrali di f_3 sul cubo, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di Gauss-Chebyshev prodotto tensoriale.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	5.11e-04	3.87e-04	3.59e-04
	lsqnonneg m. esatti	6.40e-03	7.80e-03	6.41e-03
	lsqnonneg m. QMC	1.89e-02	6.80e-03	6.30e-03
	AFP m. esatti	7.50e-02	4.78e-02	1.43e-01
	AFP m. QMC	7.50e-02	4.78e-02	1.43e-01
$n = 7$	QMC	1.89e-04	6.48e-05	3.65e-05
	lsqnonneg m. esatti	4.37e-03	2.93e-03	4.14e-03
	lsqnonneg m. QMC	4.36e-03	1.06e-03	2.91e-03
	AFP m. esatti	1.12e-02	8.26e-03	2.47e-02
	AFP m. QMC	1.12e-02	8.26e-03	2.47e-02
$n = 9$	QMC	2.02e-04	7.77e-05	4.94e-05
	lsqnonneg m. esatti	1.66e-03	1.79e-03	3.33e-05
	lsqnonneg m. QMC	5.32e-04	1.31e-03	1.26e-03
	AFP m. esatti	1.42e-03	1.28e-03	4.92e-03
	AFP m. QMC	1.41e-03	1.27e-03	4.92e-03

Tabella 6.7: Errori relativi degli integrali di f_4 sul cubo, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di Gauss-Chebyshev prodotto tensoriale.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	1.85e-04	9.91e-05	5.14e-05
	lsqnonneg m. esatti	5.62e-05	2.99e-05	5.76e-05
	lsqnonneg m. QMC	1.33e-04	7.04e-05	1.05e-04
	AFP m. esatti	2.40e-04	2.70e-04	5.12e-04
	AFP m. QMC	2.50e-04	2.60e-04	5.02e-04
$n = 7$	QMC	1.85e-04	9.91e-05	5.14e-05
	lsqnonneg m. esatti	1.85e-05	1.86e-05	2.01e-05
	lsqnonneg m. QMC	1.74e-04	1.02e-04	4.87e-05
	AFP m. esatti	1.53e-05	1.85e-05	3.20e-05
	AFP m. QMC	5.03e-06	2.87e-05	4.23e-05
$n = 9$	QMC	1.85e-04	9.91e-05	5.14e-05
	lsqnonneg m. esatti	1.05e-06	1.12e-06	7.75e-07
	lsqnonneg m. QMC	1.84e-04	1.02e-04	4.76e-05
	AFP m. esatti	1.66e-06	6.20e-07	5.83e-06
	AFP m. QMC	1.20e-05	1.09e-05	4.46e-06

Tabella 6.8: Errori relativi degli integrali di f_5 sul cubo, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti esatti/QMC, e dagli AFP con momenti esatti/QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di Gauss-Chebyshev prodotto tensoriale.

6.2 Cono

Consideriamo il cono di altezza $h = 3$ e base con raggio $r = 1$ centrata nell'origine. Le configurazioni iniziali sono costituite da punti di Halton. I momenti QMC sono stati determinati a partire da una griglia iniziale di 1500000 di punti di Halton.

Nella Figura 6.2 si possono vedere le ditribuzioni degli AFP da WAM, dei nodi prodotti da **lsqnonneg**, e degli AFP Estratti. Sia i punti selezionati da **lsqnonneg** sia gli AFP Estratti sono distribuiti in modo omogeneo su tutto il dominio, accumulandosi verso il bordo.

In Tabella 6.9 riportiamo il numero di nodi prodotti da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton. Il numero degli AFP da WAM e degli AFP Estratti è naturalmente uguale, ed è pari a $(n+1)(n+2)(n+3)/6$. I punti determinati da **lsqnonneg** risultano in numero uguale a quello degli AFP. I rapporti dei pesi denotano una presenza non irrilevante di pesi negativi, in particolare per gli AFP da WAM. Per questi ultimi, inoltre, si può vedere che i rapporti ρ tendono a crescere all'aumentare del grado n . Dalla teoria ci si attende, quindi, che i risultati ottenuti dagli AFP da WAM (che tra l'altro sono quelli che prendiamo come metro di riferimento) possano essere inficiati da fenomeni di instabilità, in particolare per valori di n elevati.

In Tabella 6.10 sono elencati i risultati della cubatura sugli AFP da WAM, che utilizziamo come valori di riferimento per il calcolo degli errori relativi compiuti (a pari grado n) dall'integrazione QMC, dalla compressione via **lsqnonneg**, e dalla cubatura sugli AFP Estratti (6.11-6.15). Si noti che per la funzione f_3 i valori integrali sono molto variabili, e che quelli delle altre funzioni (con l'eccezione di f_5) differiscono già dalla prima cifra decimale. Tutto questo è in linea

con il fatto che può esserci instabilità ad opera dei pesi negativi degli AFP da WAM. Si tenga presente, quindi, che gli errori relativi potranno essere di entità non irrilevante, in quanto calcolati rispetto ad integrali già di per sé non esatti.

La funzione f_3 è quella che dà i risultati peggiori, mentre f_5 presenta gli errori più piccoli. Questo comportamento non solo è spiegabile dai risultati in Tabella 6.10 rispettivamente poco e abbastanza esatti, ma dipende anche dall'entità delle variazioni delle due funzioni, in virtù del Teorema di Koksma-Hlawka.

Fatta eccezione per la funzione f_3 per i motivi spiegati sopra, il metodo QMC funziona abbastanza bene, pur non registrando significativi miglioramenti all'aumentare della cardinalità N della griglia di Halton.

La cubatura con `lsqnonneg` e sugli AFP Estratti risulta buona almeno quasi quanto quella QMC. In generale, per i gradi considerati, il metodo `lsqnonneg` e quello degli AFP Estratti producono risultati equivalenti, tranne che per f_5 , i cui errori ad opera degli AFP Estratti sono di molto migliori.

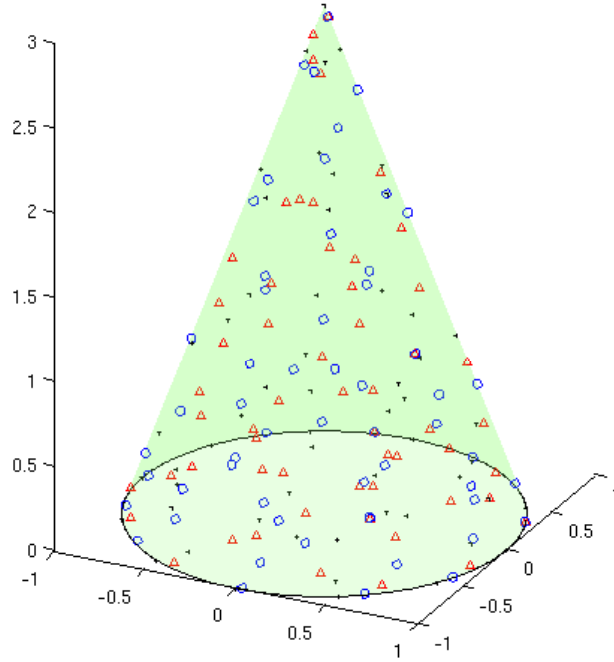


Figura 6.2: Cono con $N = 100000$, $n = 5$: distribuzione degli AFP prodotti da 3dWAM (+), dei punti estratti dalla griglia di Halton tramite `lsqnonneg` con momenti QMC (\triangle), e degli AFP (\circ).

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	3dWAM	56 (1.36)	56 (1.36)	56 (1.36)
	QMC	13104	26215	52382
	lsqnonneg m. QMC	56	56	56
	AFP m. QMC	56 (1.14)	56 (1.10)	56 (1.01)
$n = 7$	3dWAM	120 (1.66)	120 (1.66)	120 (1.66)
	QMC	13104	26215	52382
	lsqnonneg m. QMC	120	120	120
	AFP m. QMC	120 (1.15)	120 (1.16)	120 (1.03)
$n = 9$	3dWAM	220 (1.88)	220 (1.88)	220 (1.88)
	QMC	13104	26215	52382
	lsqnonneg m. QMC	220	220	220
	AFP m. QMC	220 (1.14)	220 (1.13)	220 (1.07)

Tabella 6.9: Numero dei nodi sul cono prodotti da 3dWAM, dei punti quasi-random contenuti nel dominio, dei punti estratti da lsqnonneg con momenti QMC, e degli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / \sum w_i$.

f_1	$n = 5$	9.9460474569415369e-01
	$n = 7$	9.8497107862267430e-01
	$n = 9$	9.6735495854331466e-01
f_2	$n = 5$	2.5136176567198705e+00
	$n = 7$	2.5036740694784263e+00
	$n = 9$	2.4820322060784306e+00
f_3	$n = 5$	-2.3208106151309230e-01
	$n = 7$	5.2122471262591619e-02
	$n = 9$	3.5682293956500633e-02
f_4	$n = 5$	3.0391933114362607e-01
	$n = 7$	2.9997419414872467e-01
	$n = 9$	2.9486275299536463e-01
f_5	$n = 5$	5.2974798525586637e+00
	$n = 7$	5.2986131068283555e+00
	$n = 9$	5.2982363826454826e+00

Tabella 6.10: Valori degli integrali ottenuti da 3dWAM sul cono, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	2.13e-02	2.33e-02	2.36e-02
	lsqnonneg m. QMC	3.23e-02	3.40e-02	4.30e-02
	AFP m. QMC	4.29e-02	3.37e-02	3.56e-02
$n = 7$	QMC	1.18e-02	1.37e-02	1.40e-02
	lsqnonneg m. QMC	3.70e-02	8.12e-04	6.99e-03
	AFP m. QMC	1.81e-02	7.63e-03	9.61e-03
$n = 9$	QMC	6.24e-03	4.22e-03	3.94e-03
	lsqnonneg m. QMC	8.18e-03	4.66e-03	5.29e-03
	AFP m. QMC	9.10e-03	6.18e-03	6.19e-03

Tabella 6.11: Errori relativi degli integrali di f_1 sul cono, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di **3dWAM**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	2.68e-03	4.26e-03	4.07e-03
	lsqnonneg m. QMC	1.19e-02	4.13e-03	9.77e-03
	AFP m. QMC	7.53e-03	1.09e-02	3.75e-04
$n = 7$	QMC	1.28e-03	3.03e-04	1.15e-04
	lsqnonneg m. QMC	2.49e-03	2.94e-03	2.30e-03
	AFP m. QMC	3.84e-03	3.20e-03	7.33e-03
$n = 9$	QMC	1.00e-02	8.41e-03	8.60e-03
	lsqnonneg m. QMC	1.44e-02	5.31e-04	8.67e-03
	AFP m. QMC	8.06e-03	6.24e-03	6.54e-03

Tabella 6.12: Errori relativi degli integrali di f_2 sul cono, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di **3dWAM**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	1.06e+00	1.05e+00	1.05e+00
	lsqnonneg m. QMC	1.88e+00	2.33e+00	2.90e+00
	AFP m. QMC	1.77e+00	1.59e+00	1.93e+00
$n = 7$	QMC	7.46e-01	7.74e-01	7.56e-01
	lsqnonneg m. QMC	8.97e-01	4.77e+00	4.18e-01
	AFP m. QMC	1.06e+00	2.55e+00	6.64e-01
$n = 9$	QMC	6.30e-01	6.70e-01	6.43e-01
	lsqnonneg m. QMC	2.53e-01	7.16e-01	9.65e-01
	AFP m. QMC	5.69e-01	2.84e-01	9.29e-01

Tabella 6.13: Errori relativi degli integrali di f_3 sul cono, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di **3dWAM**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	3.18e-02	3.24e-02	3.16e-02
	lsqnonneg m. QMC	6.43e-02	1.48e-02	8.94e-02
	AFP m. QMC	6.55e-02	5.97e-02	5.15e-02
$n = 7$	QMC	1.91e-02	1.97e-02	1.88e-02
	lsqnonneg m. QMC	2.14e-02	2.04e-02	3.55e-02
	AFP m. QMC	3.17e-02	2.91e-02	2.08e-02
$n = 9$	QMC	2.08e-03	2.67e-03	1.82e-03
	lsqnonneg m. QMC	4.47e-03	4.59e-03	1.23e-02
	AFP m. QMC	1.02e-02	6.72e-03	3.29e-03

Tabella 6.14: Errori relativi degli integrali di f_4 sul cono, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di **3dWAM**.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	6.25e-03	4.11e-03	2.18e-03
	lsqnonneg m. QMC	6.38e-03	4.28e-03	2.22e-03
	AFP m. QMC	1.57e-04	3.66e-04	3.64e-04
$n = 7$	QMC	6.03e-03	3.89e-03	1.96e-03
	lsqnonneg m. QMC	6.02e-03	3.93e-03	2.08e-03
	AFP m. QMC	8.44e-05	1.14e-04	7.83e-05
$n = 9$	QMC	6.10e-03	3.97e-03	2.03e-03
	lsqnonneg m. QMC	6.09e-03	3.95e-03	2.05e-03
	AFP m. QMC	1.80e-05	8.34e-06	1.58e-07

Tabella 6.15: Errori relativi degli integrali di f_5 sul cono, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di `3dWAM`.

6.3 Piramide

Consideriamo la piramide di altezza $h = 3$ e base quadrata di lato $r = 1$ centrata nell'origine. Le griglie quasi-random iniziali sono costituite da punti di Halton. I momenti QMC sono stati determinati a partire da una griglia iniziale di 1500000 di punti di Halton.

Nella Figura 6.3 sono rappresentati gli AFP da WAM, i nodi prodotti da `lsqnonneg`, e gli AFP Estratti. Sia i punti selezionati da `lsqnonneg` sia gli AFP Estratti sono distribuiti in modo omogeneo su tutto il dominio, accumulandosi verso il bordo.

La Tabella 6.16 riporta il numero di nodi prodotti da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton. Gli AFP da WAM e degli AFP Estratti sono in numero uguale, ovvero $(n+1)(n+2)(n+3)/6$, come previsto dalla teoria. I punti determinati da `lsqnonneg` risultano esattamente tanti quanti gli AFP. Anche in questo caso, i rapporti dei pesi rivelano una presenza non irrilevante di pesi negativi, in particolare per gli AFP da WAM. Dalla teoria ci si attende, quindi, che i risultati ottenuti dagli AFP da WAM (che tra l'altro sono quelli che prendiamo come integrali di riferimento) possano essere inquinati da fenomeni di instabilità.

In Tabella 6.17 sono elencati i risultati della cubatura sugli AFP da WAM, che utilizziamo come valori di riferimento per il calcolo degli errori relativi compiuti (a pari grado n) dall'integrazione QMC, dalla compressione via `lsqnonneg`, e dalla cubatura sugli AFP Estratti (Tabelle 6.18-6.22). Si noti che anche su questo dominio i valori degli integrali della funzione f_3 sono molto variabili, e che quelli delle altre funzioni (con l'eccezione di f_5) differiscono già almeno dalla prima cifra decimale. Tutto questo è spiegabile con il fatto che i pesi negativi degli AFP da WAM possono dare instabilità (cfr. i rapporti dei pesi in Tabella 6.16). Si tenga presente, quindi, che gli errori relativi potranno essere elevati, in quanto calcolati rispetto a integrali già di per sé non esatti.

La funzione f_3 dà risultati non buoni, mentre f_5 presenta errori particolarmente piccoli. Questo comportamento trova una giustificazione nei risultati della Tabella 6.17, che in un caso non sono affatto esatti mentre nell'altro lo sono

abbastanza, e nel Teorema di Koksma-Hlawka a causa delle variazioni delle due funzioni (l'una elevata, mentre l'altra ridotta).

Fatta eccezione per la funzione f_3 per i motivi spiegati sopra, il metodo QMC funziona abbastanza bene; tuttavia, a parte f_5 , non dimostra significativi miglioramenti all'aumentare della cardinalità N della griglia di Halton.

La cubatura con **lsqnonneg** e sugli AFP Estratti risulta buona almeno quasi quanto quella QMC. In generale, per i gradi considerati, il metodo **lsqnonneg** e quello degli AFP Estratti producono risultati equivalenti. Si può tuttavia notare, soprattutto per f_5 , che gli errori ad opera degli AFP Estratti sono di molto migliori per gradi alti, mentre **lsqnonneg** funziona meglio per gradi bassi e cardinalità N elevata.

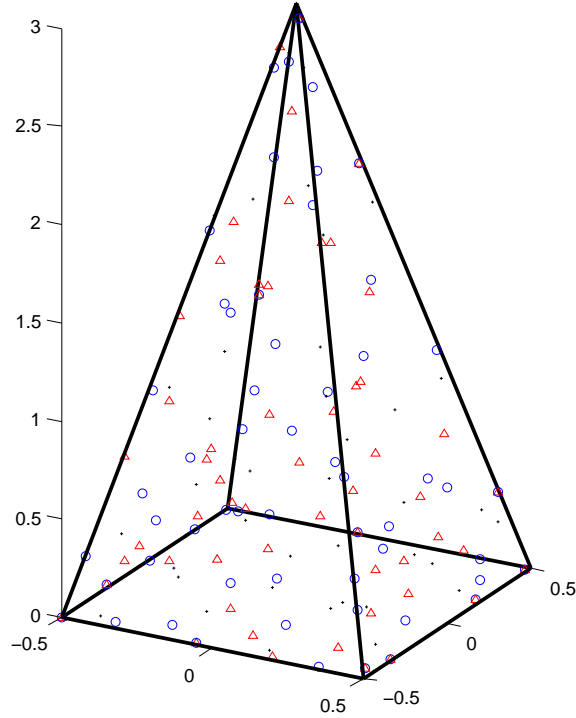


Figura 6.3: Piramide con $N = 100000$, $n = 5$: distribuzione degli AFP prodotti da 3dWAM (+), dei punti estratti dalla griglia di Halton tramite **lsqnonneg** con momenti QMC (\triangle), e degli AFP (\circ).

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	3dWAM	56 (2.81)	56 (2.81)	56 (2.81)
	QMC	16658	33341	66648
	lsqnonneg m. QMC	56	56	56
	AFP m. QMC	56 (1.81)	56 (2.92)	56 (1.73)
$n = 7$	3dWAM	120 (3.45)	120 (3.45)	120 (3.45)
	QMC	16658	33341	66648
	lsqnonneg m. QMC	120	120	120
	AFP m. QMC	120 (1.34)	120 (1.43)	120 (1.50)
$n = 9$	3dWAM	220 (2.24)	220 (2.24)	220 (2.24)
	QMC	16658	33341	66648
	lsqnonneg m. QMC	220	220	220
	AFP m. QMC	220 (1.47)	220 (1.38)	220 (1.45)

Tabella 6.16: Numero dei nodi sulla piramide prodotti da 3dWAM, dei punti quasi-random contenuti nel dominio, dei punti estratti da lsqnonneg con momenti QMC, e degli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / \sum w_i$.

f_1	$n = 5$	3.6906762842237600e-01
	$n = 7$	3.4486614522752440e-01
	$n = 9$	3.6033126901275492e-01
f_2	$n = 5$	6.9324416024630575e-01
	$n = 7$	7.0181177694456798e-01
	$n = 9$	6.9616705298783277e-01
f_3	$n = 5$	-1.0082090129946950e-01
	$n = 7$	1.0012533247858529e-02
	$n = 9$	-5.5926479668052323e-03
f_4	$n = 5$	1.3725061495833241e-01
	$n = 7$	1.5329949863979292e-01
	$n = 9$	1.4975439016886893e-01
f_5	$n = 5$	1.4419961482440700e+00
	$n = 7$	1.4428011893698325e+00
	$n = 9$	1.4427058603642284e+00

Tabella 6.17: Valori degli integrali ottenuti da 3dWAM sulla piramide, al variare della funzione integranda e del grado n .

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	2.56e-02	2.64e-02	2.63e-02
	lsqnonneg m. QMC	7.31e-02	5.86e-02	6.56e-02
	AFP m. QMC	6.49e-02	5.10e-02	3.22e-02
$n = 7$	QMC	4.28e-02	4.20e-02	4.20e-02
	lsqnonneg m. QMC	3.34e-02	4.54e-02	4.54e-02
	AFP m. QMC	4.68e-02	5.57e-02	5.48e-02
$n = 9$	QMC	1.97e-03	2.76e-03	2.74e-03
	lsqnonneg m. QMC	5.11e-03	1.17e-02	8.10e-03
	AFP m. QMC	1.08e-03	4.82e-03	1.45e-02

Tabella 6.18: Errori relativi degli integrali di f_1 sulla piramide, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di 3dWAM.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	4.26e-03	4.19e-03	4.25e-03
	lsqnonneg m. QMC	2.64e-03	7.35e-03	7.31e-03
	AFP m. QMC	4.83e-03	2.41e-02	4.13e-03
$n = 7$	QMC	8.00e-03	8.07e-03	8.01e-03
	lsqnonneg m. QMC	7.81e-03	8.83e-03	7.76e-03
	AFP m. QMC	6.81e-03	9.84e-03	6.92e-03
$n = 9$	QMC	4.70e-05	2.81e-05	3.75e-05
	lsqnonneg m. QMC	4.60e-05	6.84e-06	7.99e-04
	AFP m. QMC	1.48e-03	4.60e-04	3.07e-04

Tabella 6.19: Errori relativi degli integrali di f_2 sulla piramide, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di 3dWAM.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	9.35e-01	9.36e-01	9.34e-01
	lsqnonneg m. QMC	1.23e+00	1.16e+00	1.24e+00
	AFP m. QMC	2.22e+00	1.32e+00	8.55e-02
$n = 7$	QMC	1.65e+00	1.65e+00	1.66e+00
	lsqnonneg m. QMC	3.69e+00	1.67e+00	2.92e+00
	AFP m. QMC	3.50e-01	3.87e+00	1.32e+00
$n = 9$	QMC	1.64e-01	1.61e-01	1.87e-01
	lsqnonneg m. QMC	2.65e-01	1.55e-01	2.84e-01
	AFP m. QMC	5.10e-01	2.45e-01	4.41e-01

Tabella 6.20: Errori relativi degli integrali di f_3 sulla piramide, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di 3dWAM.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	9.28e-02	9.19e-02	9.22e-02
	lsqnonneg m. QMC	2.65e-02	4.26e-02	2.00e-02
	AFP m. QMC	1.41e-02	2.14e-01	5.10e-02
$n = 7$	QMC	2.16e-02	2.24e-02	2.21e-02
	lsqnonneg m. QMC	2.22e-02	2.06e-02	3.05e-02
	AFP m. QMC	1.52e-02	2.12e-02	2.06e-02
$n = 9$	QMC	1.53e-03	7.42e-04	1.05e-03
	lsqnonneg m. QMC	1.45e-03	2.93e-03	3.30e-05
	AFP m. QMC	8.19e-03	5.88e-04	7.03e-03

Tabella 6.21: Errori relativi degli integrali di f_4 sulla piramide, ottenuti dal metodo QMC su punti di Halton, da **lsqnonneg** con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di 3dWAM.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	1.44e-03	1.89e-03	4.63e-04
	lsqnonneg m. QMC	8.05e-04	2.28e-03	8.97e-04
	AFP m. QMC	8.22e-04	9.74e-04	1.09e-03
$n = 7$	QMC	1.99e-03	1.33e-03	9.49e-05
	lsqnonneg m. QMC	1.95e-03	1.36e-03	2.98e-05
	AFP m. QMC	2.37e-05	4.20e-05	1.07e-04
$n = 9$	QMC	1.93e-03	1.40e-03	2.88e-05
	lsqnonneg m. QMC	1.93e-03	1.41e-03	3.63e-05
	AFP m. QMC	1.34e-05	6.68e-06	2.46e-07

Tabella 6.22: Errori relativi degli integrali di f_5 sulla piramide, ottenuti dal metodo QMC su punti di Halton, da `lsqnonneg` con momenti QMC, e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto ai risultati di 3dWAM.

6.4 Dominio composito

Come dominio composito abbiamo scelto l'unione del cubo $[0, 0.75]^3$ con la sfera di raggio $r = 0.5$ centrata in $C = (0.5, 0.5, 0.5)$ (Figura 6.4). Le griglie quasirandom iniziali sono costituite da punti di Halton. I momenti QMC sono stati determinati a partire da una griglia iniziale di 1500000 di punti di Halton.

In Figura 6.5 sono rappresentati i nodi prodotti da `lsqnonneg` e gli AFP Estratti. Sia i punti selezionati da `lsqnonneg` sia gli AFP Estratti sono distribuiti in modo omogeneo su tutto il dominio, accumulandosi verso il bordo.

La Tabella 6.24 riporta il numero di nodi prodotti da ciascun metodo, al variare del grado n e della cardinalità N della griglia di Halton. Il numero degli AFP Estratti è $(n+1)(n+2)(n+3)/6$, mentre i punti determinati da `lsqnonneg` risultano essere altrettanti o di poco inferiori (per $n = 9$, $N = 200000$). I rapporti dei pesi degli AFP Estratti rivelano una presenza non trascurabile di pesi negativi, in particolare per il grado $n = 5$. Dalla teoria ci si attende, quindi, che per i risultati ottenuti dagli AFP Estratti possa esserci un po' di instabilità.

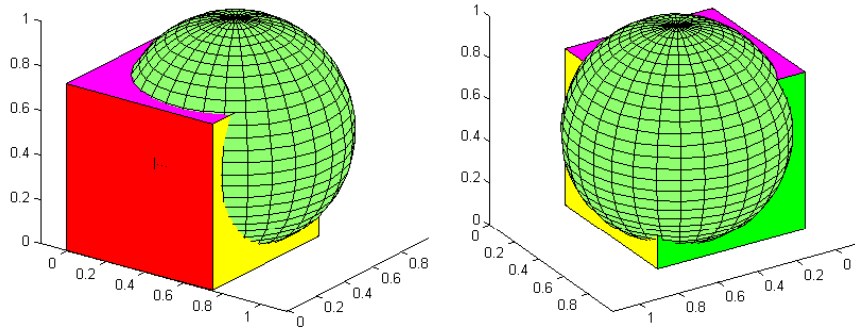


Figura 6.4: Dominio composito dato dall'unione di un cubo con una sfera.

In Tabella 6.23 sono elencati i risultati dell'integrazione QMC, che utilizziamo come valori di riferimento per il calcolo degli errori relativi compiuti (a pari cardinalità N) dalla compressione via `lsqnonneg` e dalla cubatura sugli AFP Estratti (6.25-6.29). I valori degli integrali della funzione f_5 presentano qualche sbavatura, mentre quelli delle altre funzioni sono abbastanza uniformi.

Per i gradi qui considerati, gli errori commessi da `lsqnonneg` e dagli AFP Estratti risultano piuttosto buoni. A parte qualche raro caso, il metodo `lsqnonneg` si dimostra più efficiente di quello degli AFP Estratti.

f_1	$N = 50000$	1.6884821698971897e-01
	$N = 100000$	1.6882130540933493e-01
	$N = 200000$	1.6876617653812614e-01
f_2	$N = 50000$	2.3492767985452040e-01
	$N = 100000$	2.3511617659489875e-01
	$N = 200000$	2.3501735927865819e-01
f_3	$N = 50000$	4.3754580960267242e-02
	$N = 100000$	4.3785859610183414e-02
	$N = 200000$	4.3767157038649390e-02
f_4	$N = 50000$	6.9170138513218918e-02
	$N = 100000$	6.9159365862525940e-02
	$N = 200000$	6.9141699988675143e-02
f_5	$N = 50000$	6.9170138513218918e-02
	$N = 100000$	5.0818301176270708e-01
	$N = 200000$	5.0763702196562766e-01

Tabella 6.23: Valori degli integrali sul dominio composito ottenuti dal metodo QMC usando punti di Halton, al variare della funzione integranda e della cardinalità N della griglia iniziale sul rettangolo circoscritto al dominio.

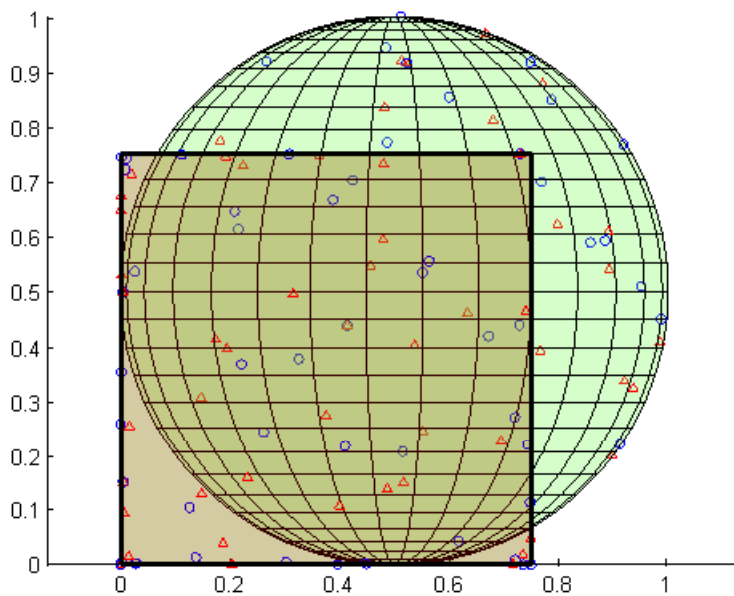
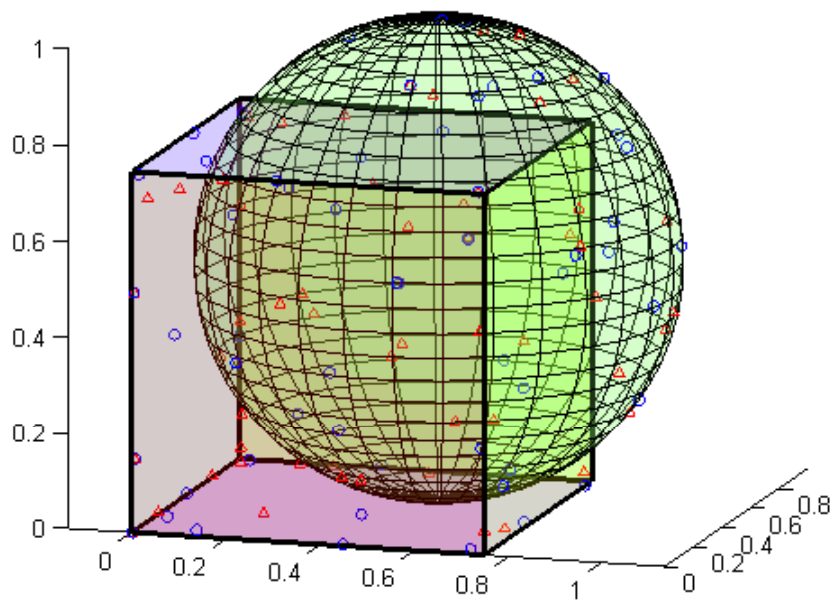


Figura 6.5: Dominio composto con $N = 100000$, $n = 5$: distribuzione dei punti estratti dalla griglia di Halton tramite `lsqnonneg` con momenti QMC (\triangle), e degli AFP (\circ).

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	QMC	32212	64431	128793
	lsqnonneg m. QMC	56	56	56
	AFP m. QMC	56 (1.47)	56 (1.62)	56 (1.28)
$n = 7$	QMC	32212	64431	128793
	lsqnonneg m. QMC	120	120	120
	AFP m. QMC	120 (1.48)	120 (1.15)	120 (1.18)
$n = 9$	QMC	32212	64431	128793
	lsqnonneg m. QMC	220	220	216
	AFP m. QMC	220 (1.27)	220 (1.24)	220 (1.26)

Tabella 6.24: Numero dei punti quasi-random contenuti nel dominio composito, dei punti estratti da **lsqnonneg** con momenti QMC, e degli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Per gli insiemi di nodi che possono avere pesi negativi, sono stati riportati tra parentesi i rapporti dei pesi $\rho = \sum |w_i| / |\sum w_i|$.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	lsqnonneg m. QMC	1.97e-02	3.70e-02	4.58e-03
	AFP m. QMC	1.99e-02	5.11e-02	1.33e-02
$n = 7$	lsqnonneg m. QMC	5.49e-04	4.62e-03	3.43e-03
	AFP m. QMC	2.64e-02	3.92e-03	1.07e-02
$n = 9$	lsqnonneg m. QMC	8.34e-05	3.01e-03	7.80e-03
	AFP m. QMC	2.19e-03	2.25e-04	2.82e-05

Tabella 6.25: Errori relativi degli integrali di f_1 sul dominio composito, ottenuti da **lsqnonneg** e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto al metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	lsqnonneg m. QMC	7.26e-03	1.26e-02	8.00e-03
	AFP m. QMC	2.45e-02	1.54e-02	3.97e-02
$n = 7$	lsqnonneg m. QMC	4.44e-03	9.43e-03	5.81e-03
	AFP m. QMC	8.66e-04	8.07e-03	3.45e-03
$n = 9$	lsqnonneg m. QMC	2.39e-03	4.05e-03	1.92e-03
	AFP m. QMC	4.76e-03	3.59e-03	1.65e-03

Tabella 6.26: Errori relativi degli integrali di f_2 sul dominio composito, ottenuti da **lsqnonneg** e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto al metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	lsqnonneg m. QMC	5.98e-02	9.65e-02	6.82e-02
	AFP m. QMC	1.02e-01	7.04e-03	2.76e-02
$n = 7$	lsqnonneg m. QMC	2.88e-03	8.01e-03	8.56e-04
	AFP m. QMC	1.19e-03	6.12e-04	2.05e-03
$n = 9$	lsqnonneg m. QMC	2.91e-05	2.67e-05	2.00e-04
	AFP m. QMC	1.07e-03	2.06e-03	1.69e-03

Tabella 6.27: Errori relativi degli integrali di f_3 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto al metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	lsqnonneg m. QMC	1.32e-02	5.90e-03	5.74e-03
	AFP m. QMC	2.76e-02	4.31e-03	3.77e-03
$n = 7$	lsqnonneg m. QMC	9.98e-04	1.60e-03	2.41e-05
	AFP m. QMC	4.13e-03	6.11e-03	1.27e-03
$n = 9$	lsqnonneg m. QMC	8.56e-04	1.37e-03	6.65e-04
	AFP m. QMC	6.40e-04	5.92e-04	8.25e-04

Tabella 6.28: Errori relativi degli integrali di f_4 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto al metodo QMC.

	Metodo	$N = 50000$	$N = 100000$	$N = 200000$
$n = 5$	lsqnonneg m. QMC	7.68e-06	1.22e-05	2.28e-05
	AFP m. QMC	4.06e-05	7.87e-04	3.58e-04
$n = 7$	lsqnonneg m. QMC	1.26e-05	3.17e-06	8.34e-06
	AFP m. QMC	1.16e-04	7.11e-04	3.70e-04
$n = 9$	lsqnonneg m. QMC	1.73e-06	2.12e-06	2.15e-06
	AFP m. QMC	1.19e-04	6.99e-04	3.77e-04

Tabella 6.29: Errori relativi degli integrali di f_5 sul dominio composito, ottenuti da `lsqnonneg` e dagli AFP con momenti QMC, al variare del grado n e della cardinalità N della griglia di Halton iniziale. Gli errori sono stati calcolati rispetto al metodo QMC.

Conclusioni

In questa tesi è emerso che la cubatura sui punti estratti da `lsqnonneg` e da `approxfek` da griglie quasi-random (es. Halton, Sobol) fornisce in generale risultati apprezzabili. In particolare, utilizzando `lsqnonneg` si ottengono formule di cubatura con un numero di punti tipicamente uguale (ma talvolta anche inferiore) a quello degli AFP, che è pari alla dimensione dello spazio polinomiale $\dim(\mathbb{P}_n^d) = \binom{n+d}{d}$, dove n è il grado polinomiale. I tempi di esecuzione di `lsqnonneg` sono di gran lunga minori di `approxfek` a causa dell'elevato numero di punti da cui si va ad estrarre; tuttavia, per ogni dato dominio e grado n , queste funzioni ne determinano una volta per tutte i nodi e i pesi, che quindi non devono essere più ricalcolati. Si è visto che le formule di quadratura via `lsqnonneg` forniscono risultati in linea di massima equivalenti a quelli degli AFP. Entrambi i metodi con momenti esatti si sono dimostrati più efficienti dell'integrazione quasi-Monte Carlo per gradi superiori a 10 o quantomeno a 20, eccetto che per le funzioni con variazione grande sul dominio, per le quali peraltro anche l'integrazione quasi-Monte Carlo e i metodi con grado algebrico esatto danno risultati insoddisfacenti. In generale, quindi, `lsqnonneg` e gli AFP con momenti esatti forniscono formule di cubatura più precise dell'integrazione quasi-Monte Carlo, e con un numero di nodi di molto inferiore a quello dei metodi con grado algebrico esatto qui presentati.

Le formule di cubatura ottenute estraendo i nodi da punti quasi-random e calcolando i momenti con quasi-Monte Carlo, hanno in generale performance paragonabili a QMC. Questo legittima l'uso dei momenti quasi-Monte Carlo nei casi in cui non si dispone dei momenti esatti, come si è fatto per i domini compositi. In tal modo, per i domini con geometrie difficili da trattare, si hanno due alternative al metodo di integrazione quasi-Monte Carlo, che forniscono buoni risultati con un numero di nodi estremamente ridotto.

Nel caso trivariato, rimane aperta la questione di come superare l'impasse della complessità numerica. In questo lavoro, infatti, si è riusciti a condurre esperimenti su domini 3D, usando un laptop con le caratteristiche precedentemente descritte, solo fino al grado 9. Per gradi più elevati una possibile soluzione consiste nell'effettuare la compressione su calcolatori più potenti.

Osserviamo che la funzione Matlab `lsqnonneg` può essere applicata ad altri insiemi di punti da cui estrarre i nodi, come ad esempio le WAMs, oppure su nodi precedentemente selezionati da altri metodi di cubatura. Un approccio in questo senso è stato sperimentato con successo in [58] su insiemi di nodi con cardinalità abbastanza elevata, prodotti da metodi di cubatura con grado algebrico esatto. Siamo convinti che estrarre i nodi di cubatura da WAMs abbrevierà di molto i tempi di calcolo; questo sarà un futuro lavoro.

Bibliografia

- [1] A. Antonov e V. M. Saleev, *An economic method of computing LP_τ -sequences*, Comput. Maths. Math. Phys., 19 (1980), 252–256.
- [2] R. Berman e S. Boucksom, *Equidistribution of Fekete Points on Complex Manifolds*, preprint disponibile alla pagina web <http://arxiv.org/abs/0807.0035>.
- [3] L. Bos, J.P. Calvi, N. Levenberg, A. Sommariva e M. Vianello, *Geometric Weakly Admissible Meshes, Discrete Least Squares Approximations and Approximate Fekete Points*, Math. Comp. 80 (2011), 1601-1621.
- [4] L. Bos, S. De Marchi, A. Sommariva e M. Vianello, *Computing multivariate Fekete and Leja points by numerical linear algebra*, SIAM J. Numer. Anal. 48 (2010), 1984-1999.
- [5] L. Bos, S. De Marchi, A. Sommariva e M. Vianello, *Weakly Admissible Meshes and Discrete Extremal Sets*, Numer. Math. Theory Methods Appl. 4 (2011), 1-12.
- [6] L. Bos e M. Vianello, *Subperiodic trigonometric interpolation and quadrature*, Appl. Math. Comput. 218 (2012), 10630-10638.
- [7] L. Bos e M. Vianello, *trigquad.m: Matlab function for subperiodic trigonometric quadrature*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>.
- [8] P. Bratley e B. L. Fox (1988) *Algorithm 659: Implementing Sobols quasi-random sequence generator*, ACM Trans. Math. Software 14 (1988), 88–100.
- [9] R. E. Caflish, *Monte Carlo and quasi-Monte Carlo methods*, Acta Numerica 7 (1998), 1-49.
- [10] M. Caliari, S. De Marchi, R. Montagna e M. Vianello, *Hyper2d: a numerical code for hyperinterpolation on rectangles*, Appl. Math. Comput. 183 (2006), 1138-1147.
- [11] M. Caliari, S. De Marchi, R. Montagna e M. Vianello, *XuPad2D: Matlab interface for hyperinterpolation at Xu points and interpolation at Padua points on rectangles*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>.

- [12] M. Caliari, S. De Marchi, A. Sommariva e M. Vianello, *Padua2DM: a Matlab/Octave code for interpolation and cubature at the Padua points*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAsoft.html>.
- [13] M. Caliari, S. De Marchi e M. Vianello, *Padua2DM: fast interpolation and cubature at the Padua points in Matlab/Octave*, Numer. Algorithms 56 (2011), 45-60.
- [14] R. Cools, *An Encyclopaedia of Cubature Formulas*, J. Complexity 19 (2003), 445-453.
- [15] R. Cools, *Monomial cubature rules since "Stroud": a compilation - Part 2*, J. Comput. Appl. Math. 112(1-2) (1999), 21-27.
- [16] R. Cools e E. Goor, *Encyclopaedia of Cubature Formulas*, sito web <http://nines.cs.kuleuven.be/research/ecf/ecf.html>.
- [17] R. Cools e A. Haegemans, *Another step forward in searching for cubature formulae with a minimal number of knots for the square*, Computing 40 (1988), 139-146.
- [18] R. Cools e K. J. Kim, *A survey of known and new cubature formulas for the unit disk*, Korean J. Comput. Appl. Math. 7 (2000), 477-485.
- [19] R. Cools e K. J. Kim, *Rotation Invariant Cubature Formulas over the n-Dimensional Unit Cube*, J. Comput. Appl. Math. 132 (2001), 15-32.
- [20] R. Cools e J.N. Lyness, *A survey of numerical cubature over triangles*, in W. Gautschi editor, *Mathematics of computation 1943-1993: A half-century of computational mathematics*, Proceedings of Symposia in Applied Mathematics, Vol. 48, pages 127-150, American Mathematical Society, Providence, Rhode Island (1994).
- [21] R. Cools e P. Rabinowitz, *Monomial cubature rules since "Stroud": a compilation*, J. Comput. Appl. Math. 48 (1993), 309-326.
- [22] R. Cools e I. H. Sloan, *Minimal cubature formulae of trigonometric degree* Math. Comp. 65 (1996), 1583-1600.
- [23] R. Cools e P. Verlinden, *The algebraic construction of a minimal cubature formula of degree 11 for the square*, M.V. Noskov, editor, *Cubature Formulas and their Applications (Russian)*, Krasnoyarsk (1994), 13-23.
- [24] G. Da Fies, A. Sommariva e M. Vianello, *Algebraic cubature by linear blending of elliptical arcs*, Appl. Numer. Math. 74 (2013), 49-61.
- [25] G. Da Fies, A. Sommariva e M. Vianello, *gqellblend.m: Matlab function for product Gaussian quadrature by linear blending of elliptical arcs*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAsoft.html>.
- [26] G. Da Fies, A. Sommariva e M. Vianello, *trigauss.m: Matlab function for subperiodic trigonometric Gaussian quadrature*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAsoft.html>.

- [27] G. Da Fies e M. Vianello, *Algebraic cubature on planar lenses and bubbles*, Dolomites Res. Notes Approx. DRNA 5 (2012), 7-12.
- [28] G. Da Fies e M. Vianello, `gqdbubble.m`, `gqlens.m`, `gqlune.m` : *Matlab functions for product Gaussian quadrature on regions generated by two arbitrary overlapping disks: double bubbles (union), lenses (intersection), lunes (difference)*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>.
- [29] G. Da Fies e M. Vianello, *Trigonometric Gaussian quadrature on subintervals of the period*, Electron. Trans. Numer. Anal. 39 (2012), 102-112.
- [30] G. Da Fies e M. Vianello, `gqcircsect.m`, `gqcircsegm`, `gqcirczone.m`, `gqsymmlens.m`: *Matlab functions for product Gaussian quadrature on disk sections: circular sectors, segments, zones, symmetric lenses*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>
- [31] S. De Marchi, M. Marchioro e A. Sommariva, *Polynomial approximation and cubature at approximate Fekete and Leja points of the cylinder*, Appl. Math. Comput. Vol. 218 (2012), 10617-10629.
- [32] S. De Marchi e M. Vianello, *Hyper3: a Matlab code for cubature and hyperinterpolation in the 3-cube*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>.
- [33] S. De Marchi e M. Vianello, *Polynomial approximation on pyramids, cones and solids of rotation*, Dolomites Res. Notes Approx. DRNA 6 (2013), 20-26.
- [34] S. De Marchi e M. Vianello, *3dWAM: Matlab functions to compute weakly admissible meshes of 3-dimensional cones and solids of rotation*, disponibile alla pagina web <http://www.math.unipd.it/marcov/CAAssoft.html>.
- [35] S. De Marchi, M. Vianello e Y. Xu, *New cubature formulae and hyperinterpolation in three variables*, BIT Numerical Mathematics 49 (2009), 55-73.
- [36] G. Fasshauer, *Meshfree Approximation Methods with Matlab*, Interdisciplinary Mathematical Sciences, Vol. 6 World Scientific Publishers, Singapore (2007).
- [37] M. Festa e A. Sommariva, *Computing almost minimal formulas on the square*, J. Comput. Appl. Math. 236 (2012), 4296-4302.
- [38] G. B. Folland, *Real analysis - Modern techniques and their applications*, Wiley, 1999.
- [39] M. Gentile, A. Sommariva e M. Vianello, *GeorectQ: Matlab codes to compute algebraic quadrature formulas on geographic rectangles*, disponibile alla pagina web <http://www.math.unipd.it/alvise/papers.html>.

- [40] M. Gentile, A. Sommariva e M. Vianello, *Polynomial approximation and quadrature on geographic rectangles*, submitted, preprint disponibile alla pagina web <http://www.math.unipd.it/~alvise/papers.html>
- [41] K. Hesse, I. H. Sloan e R. S. Womersley, *Numerical integration on the sphere*, W. Freeden, M. Z. Nashed, T. Sonar (eds), Handbook of Geomathematics, Springer (2010), 1185–1219.
- [42] S. Joe e F. Kuo, *Notes on generating Sobol Sequences*, web.maths.unsw.edu.au/~sobol/joe-kuo-notes.pdf (2008).
- [43] A. Klenke, *Probability Theory: A Comprehensive Course*, Springer (2007).
- [44] L. Kuipers e H. Niederreiter, *Uniform Distribution of Sequences*, John Wiley, New York (1974).
- [45] C.L. Lawson e R.J. Hanson, *Solving Least Squares Problems*, Prentice-Hall (1974), Chapter 23, 160-165.
- [46] L. Mezzalira, A. Sommariva e M. Vianello, *Cubatura adattativa su toro e sfera*, tesi di laurea magistrale in matematica, A.A. 2012-2013, disponibile alla pagina web http://www.math.unipd.it/~alvise/TESI_STUDENTI/MEZZALIRA_2013/MEZZALIRA_2013.pdf.
- [47] S. E. Mousavi, H. Xiao e N. Sukumar, *Generalized Gaussian quadrature rules on arbitrary polygons*, Int. J. Numer. Meth. Engng, 82 (2010), 99–113.
- [48] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia (1992).
- [49] A. Punzi, A. Sommariva e M. Vianello, *GreenDisk: Matlab code for meshless cubature over disks and annular sectors by Thin-Plate Splines and Green's formula*, disponibile alla pagina web <http://www.math.unipd.it/~marcov/CAAsoft.html>.
- [50] A. Punzi, A. Sommariva e M. Vianello, *Meshless cubature over the disk by Thin-Plate Splines*, J. Comput. Appl. Math. 221 (2008), 430-436.
- [51] G. Santin, A. Sommariva and M. Vianello, *An algebraic cubature formula on curvilinear polygons*, Appl. Math. Comput. 217 (2011), 10003-10015.
- [52] L. F. Shampine, *Matlab program for quadrature in 2D*, Applied Mathematics and Computation - AMC, vol. 202, no. 1 (2008), 266-274.
- [53] L. F. Shampine, *Quadrature in 2D*, codice Matlab della routine TwoD disponibile alla pagina web <http://faculty.smu.edu/shampine/current.html>
- [54] I. H. Sloan e R. Womersley, *Constructive polynomial approximation on the sphere*, J. Approx. Theory 103 (2000), 91–118.
- [55] I. M. Sobol, *On the distribution of points in a cube and the approximate evaluation of integrals*, USSR Comput. Maths. Math. Phys. 7 (1967), 86-112.

- [56] A. Sommariva e M. Vianello, **approxfek**: *Matlab function to compute approximate Fekete and Leja points from a 2d or 3d mesh/cloud*, disponibile alla pagina web
<http://www.math.unipd.it/marcov/CAAsoft.html>
- [57] A. Sommariva e M. Vianello, **ChebfunGauss**: *Matlab code for Gauss-Green cubature by the Chebfun package*, disponibile alla pagina web
<http://www.math.unipd.it/marcov/CAAsoft.html>.
- [58] A. Sommariva e M. Vianello, *Compression of multivariate discrete measures and applications*, sottomesso a pubblicazione (2014).
- [59] A. Sommariva e M. Vianello, *Computing approximate Fekete points by QR factorizations of Vandermonde matrices*, *Comput. Math. Appl.* 57 (2009), 1324-1336.
- [60] A. Sommariva e M. Vianello, *Gauss-Green cubature and moment computation over arbitrary geometries*, *J. Comput. Appl. Math.* 231 (2009), 886-896.
- [61] A. Sommariva e M. Vianello, **Polygauss**: *Matlab code for product Gauss cubature over polygons*, software disponibile alla pagina web
<http://www.math.unipd.it/marcov/CAAsoft.html>
- [62] A. Sommariva e M. Vianello, *Product Gauss cubature over polygons based on Green's integration formula*, *BIT Numerical Mathematics* 47 (2007), 441-453.
- [63] A. Sommariva e M. Vianello, **SplineGauss**: *Matlab code for Gauss-Green cubature over spline curvilinear polygons*, disponibile alla pagina web
<http://www.math.unipd.it/marcov/CAAsoft.html>.
- [64] A.H. Stroud, *Approximate calculation of multiple integrals*, Prentice-Hall, Englewood Cliffs, N.J. (1971).
- [65] The MathWorks, *Documentation Center*, descrizione della function **dblquad** disponibile alla pagina web
<http://www.mathworks.it/it/help/matlab/ref/dblquad.html>
- [66] The MathWorks, *Documentation Center*, descrizione della function **integral2** disponibile alla pagina web
<http://www.mathworks.it/it/help/matlab/ref/integral2.html>
- [67] The MathWorks, *Documentation Center*, descrizione della function **lsqnonneg** disponibile alla pagina web
<http://www.mathworks.it/it/help/optim/ug/lsqnonneg.html>.
- [68] The MathWorks, *MATLAB documentation set*, versione 2010a, disponibile alla pagina web <http://www.mathworks.com>.
- [69] M. G. Trimbis e R. T. Trimbis, *Adaptive cubatures on triangle and spatial data structures*, *Proceedings NAAT* (2006), 401-409.